Técnicas de análisis para la mejora y predicción del rendimiento académico

Marco Antonio Cuevas Redondo
Marta Estévez Bravo

Trabajo de Fin de Grado en el Grado en Ingeniería Informática

Universidad Complutense de Madrid
Facultad de Informática

Junio 2017

Directores:

Pablo Moreno Ger

Antonio Alejandro Sánchez Ruíz-Granados

Autorización

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria como el código, la documentación y/o el prototipo desarrollado. Difundiéndose, el presente proyecto, bajo licencia GNU GPL versión 3. Así mismo, el contenido multimedia generado por los autores, se difunde bajo licencia Creative Commons BY SA versión 3.0 (CC BY-SA 3.0).

Marco Antonio Cuevas Redondo

Marta Estévez Bravo

Agradecimientos

Queremos dedicar este proyecto a Pablo y a Antonio por guiarnos a lo largo del camino que ha supuesto este TFG, a Ana por ayudarnos con nuestra carencia de conocimiento en inglés y a todos aquellos que nos han aguantado y apoyado durante toda la carrera.

Resumen

Desde hace varias décadas, profesores y pedagogos han mostrado un fuerte interés en la posibilidad de predecir cuál va a ser el desempeño académico de los estudiantes. Por ello, se han realizado multitud de estudios teniendo en cuenta variables socioeconómicas y de carácter académico. Sin embargo, apenas hay herramientas que permitan a educadores y alumnos hacer una predicción real de cuáles pueden ser los resultados a final de curso.

Por otro lado, los docentes pueden encontrarse con mucha información de cada uno de sus alumnos que sea difícil de gestionar o analizar de forma conjunta. Hay herramientas que les permiten hacer estadísticas simples con facilidad, pero pocas que permitan combinar más de tres variables distintas.

Este proyecto pretende hacer una primera aproximación a estos problemas. El objetivo es desarrollar una herramienta que permita al profesor analizar el desempeño actual de sus alumnos, así como tener una visión previa de cuáles pueden ser los resultados a final de curso.

Para hacer esto, el proyecto consta de tres funcionalidades principales que intentarán ayudar a los profesores a hacer un seguimiento de sus clases:

La primera se trata de un analizador estadístico, cuya función es dar forma a todos los datos de los que disponga el sistema para hacerlos más comprensibles.

La segunda es un agrupador de información. El objetivo de esta funcionalidad es proporcionar al profesor grupos de alumnos que presentan un determinado patrón de comportamiento.

Por último, está la funcionalidad de predicción. Esta parte se encarga de predecir la potencial nota de cada estudiante basándose en experiencia adquirida de cursos anteriores.

Abstract

For several decades, teachers and pedagogues have shown an interest in the possibility of predicting their students future performance. For this reason there have been several studies taking into account socioeconomic and academic variables. However, there are few tools that allow educators and students to make a real prediction of their likeky final marks.

What's more, teachers can find themselves with a lot of information about their pupils, but this info could be difficult to process or analyze as a whole. There are tools that allow them to do simple statistics easily, but very few of them allow combinations of more than three different variables.

This project aims to make a first approach to these problems. Our objective is developing a tool that anables the teacher to analyze their pupils present performance, as well as have a preliminary view of their probable results at the end of the school year.

To achieve this, the project consist of three main functionalities with the aim of helping teachers to track their classes:

The first one is an statistical analyzer. Its function is giving shape to all the data avaliable to the system, to make them easier to understand.

The second one is an information cluster. Its function is showing the teacher groups of students with a given behavioral pattern.

The last functionality is prediction. This section function is predicting the potential grade of any student based on the experience from previous years.

Palabras clave

- Análisis estadístico
- Predicción
- Clustering
- Calificaciones
- Aprendizaje automático
- Evalución contínua

Keywords

- Statistical analysis
- Prediction
- Clustering
- Grades
- Machine Learning
- Continuous assessment

Índice

Tabla de ilustraciones

Capítulo 1: Introducción

Capítulo 1: Introducción

1.1 - Motivación:

El presente proyecto permite al profesor realizar un seguimiento de la evolución de sus alumnos, buscar grupos de alumnos con conductas similares y predecir la nota final que podría tener cada uno de ellos. A la hora de predecir la nota del alumno, el sistema se basa en un conjunto de datos recopilados en cursos anteriores. De esta forma, el objetivo de nuestro sistema consiste en facilitar la labor docente y aprovechar la ingente cantidad de datos recopilados de años anteriores.

En la actualidad, podemos observar que existen distintos tipos de proyectos que se sirven de técnicas estadísticas o de técnicas basadas en aprendizaje automático para dar a los usuarios una visión global del sistema y predecir situaciones o datos que les ayuden en la toma de decisiones. Un ejemplo serían las aplicaciones de análisis de los sistemas financieros, estas muestran gráficas estadísticas y predicen posibles resultados económicos. Basándose en estos datos, un equipo directivo sería capaz de tomar mejores decisiones. Esto causa un gran impacto y, con el tiempo, ha pasado a ser una parte fundamental del sector.

Sin embargo, estas técnicas no sólo se pueden aplicar al sector económico. También se utilizan en proyectos tales como: motores de búsqueda, antivirus, aplicaciones de comprensión de textos o reconocimiento de imágenes, entre otros.

A pesar de lo bien que han funcionado este tipo de técnicas en dichos sectores, a día de hoy existen pocas líneas de trabajo que utilicen estas técnicas enfocadas al ambiente académico.

Si observamos el contexto social del ambiente académico que acabamos de mencionar, se puede ver que durante mucho tiempo se han realizado multitud de estudios que investigan las causas y los factores que llevan a los estudiantes al abandono de sus respectivas titulaciones. Tras dichas investigaciones, podemos destacar algunos factores tales como los problemas socioeconómicos del individuo, el desconocimiento de la titulación antes de ingresar a la misma o incluso, la falta de conocimientos previos adecuados.

No obstante, uno de los factores con mayor índice de recurrencia que podemos encontrar es la falta de un seguimiento adecuado del curso lectivo que suele deberse a distintas causas. Entre ellas destacamos:

- El desconocimiento previo del ritmo o particularidades de una asignatura concreta por parte del alumno. Esto desemboca en una mala gestión del tiempo.

- El grado de complejidad que se le presenta a un profesor a la hora de realizar un seguimiento académico de sus alumnos.

Todas estas causas y factores, así como el contexto social y tecnológico en el que nos hemos situado, nos llevan a la conclusión de que el uso de un sistema estadístico/predictivo aplicado al seguimiento académico mejoraría el rendimiento por ambas partes.

Los resultados académicos de un alumno corresponden con una serie de variables de tipo socioeconómico y un patrón de comportamiento. En la mayoría de los casos, las variables previamente mencionadas suelen ser desconocidas para el profesor, pero la actuación en el aula no. El comportamiento y actitud de un alumno hacia la asignatura puede preverse basándose en experiencias anteriores, hasta el punto de llegar a conocer, grosso modo, cuál puede ser la calificación final de un estudiante.

Como veremos a continuación, existe un gran interés en llevar estas metodologías de análisis al ámbito académico.

1.1.1 Acceso a la universidad

A lo largo de los años, se han realizado diversos estudios que buscan utilizar la idea de predecir el progreso académico para modificar los criterios de acceso a la universidad en España. Esto busca sustituir o devaluar las Pruebas de Acceso a la Universidad (PAU), que realmente no muestran si el alumno va a rendir correctamente durante su tiempo de formación superior. Esta idea se muestra, por ejemplo, en "*La predicción del rendimiento como criterio para el ingreso en la universidad*" [1], que propone utilizar las notas de los últimos tres cursos de enseñanza media baremadas en función del centro académico de origen y al que se desea ingresar para predecir, mediante regresión múltiple, si el alumno aprovechará su tiempo o no.

Desde 2003, las universidades inauguradas antes de 1981 en Chile utilizan los resultados de una *Prueba de Selección Universitaria* (PSU) ponderados con la media de las *Notas de Enseñanza Media* (NEM). Con esto generan un ranking y predicen cuál va a ser el futuro rendimiento académico del alumno. Así deciden si dicho estudiante puede acceder a la universidad que le interesa. Como se muestra en el estudio "*Predicción de notas en Derecho de la Universidad de Chile: ¿sirve el ranking?*" [2], el sistema predictivo funciona como se espera, aunque tiene en cuenta las variables socioeconómicas más de lo que aparenta inicialmente.

En la India, todos los años se realizan exámenes de acceso a la universidad que marcan si un estudiante puede entrar en un determinado centro universitario. Estas pruebas suponen un fuerte condicionante en cuanto a la calidad de la enseñanza que recibirán en un futuro. Debido a la tardanza de los resultados, hay multitud de predictores online que permiten al alumno hacerse una idea de su futuro más cercano. Suelen pedir la fecha de realización del examen y la nota esperada en el mismo, aunque algunos también solicitan un código de examen. A diferencia de lo que sucede en Chile, estas predicciones no son un condicionante para poder ir a la universidad [3] [4].

1.1.2 En el aula

En el año 2000, el estudio "*La predicción del rendimiento académico: regresión lineal versus regresión logística*" [5] analiza brevemente trabajos previos y compara la regresión lineal múltiple con la regresión lógica. Utilizando calificaciones, asistencia a clase, participación en la misma y un cuestionario con diferentes preguntas sobre la situación del estudiante antes de comenzar sus estudios universitarios. El autor concluye que el mejor tipo de técnica predictiva para realizar la tarea es la regresión lógica.

También existe *Front Row Education* [6], una herramienta que permite al profesor analizar el avance de sus alumnos sobre los conocimientos que les imparte y darles material extra para trabajar. El docente tiene toda la información que necesita en formato de tabla y en gráficas, tanto por grupo como por alumno. Además de ser capaz de cambiar el idioma en el que el alumno recibe el material extra y modificar en qué estándar se evalúan.

1.2 - Objetivos:

El proyecto que se presenta en este documento, tiene como objetivo la creación de un prototipo de una herramienta que permita al profesor hacer un seguimiento de su clase. Este seguimiento se hará a través de visualización de estadísticas, agrupamiento de alumnos con conductas similares y predicción de notas. Aunque el desarrollo se centrará en la asignatura concreta de Desarrollo de Sistemas Interactivos (DSI), el sistema constará de una arquitectura modular que permita ampliarse con facilidad. De esta forma, en el apartado trabajo futuro podrá adaptarse a nuevas asignaturas y añadir nuevas funcionalidades.

Para esto, el proyecto se basará en la visualización de los datos en un formato agradable para el usuario, la creación de modelos de conducta de los alumnos que cursan una determinada asignatura y el uso de técnicas de aprendizaje automático para intentar predecir la nota final de alumno.

En esta versión del proyecto, nos centramos en el desarrollo de la arquitectura interna del sistema y en la funcionalidad dedicada al profesor. A pesar de esto, pondremos especial precaución para que, en trabajo futuro, sea fácil dar soporte a la funcionalidad del alumno.

Para la implementación del sistema se han fijado unos objetivos que pasamos a detallar a continuación:

- La herramienta debe ser capaz de obtener una serie de datos a raíz de una serie de ficheros excel o con extensión '.csv' con un formato definido previamente y transformarlos en información que sea de fácil manejo y de relevancia para el sistema.

- El sistema deberá proporcionar una sección importante dedicada a la visualización de gráficas estadísticas. Se hará a partir de los datos que el profesor proporcione al sistema sobre el curso que imparte.

- Detección de grupos de alumnos con conductas similares y visualización de dichos grupos a través de gráficas. Se mostrará, además, una lista que indique en qué grupo se encuentra cada alumno.

- Se proporcionará un sistema de predicción sencillo. Éste deberá mostrar, para cada alumno de la clase, una predicción de su posible nota final basada en su rendimiento en la asignatura y comparándolo con el rendimiento de alumnos de cursos anteriores. Esta nota estará entre uno de los siguientes valores: suspenso, aprobado, notable y sobresaliente.

Este último punto que acabamos de mencionar, tiene como objetivo investigar la viabilidad de los algoritmos predictivos en este entorno y, aunque el proyecto constará de un sistema de predicción sencillo, esta parte del proyecto quedará para estudio y mejora en trabajo futuro.

Con estos objetivos cumplidos, el sistema ofrece una vía de seguimiento del curso lectivo que pretende obtener una mejora en el rendimiento del mismo.

1.3 - Estructura de la Memoria:

El Capítulo 3 explica cómo se ha dividido el tiempo del proyecto en etapas y qué trabajos se han realizado en cada una de ellas. Además, se hace un repaso a todas las herramientas y librerías utilizadas para el desarrollo del trabajo.

En el Capítulo 4, por otro lado, se hace una descripción general de nuestro sistema desde el punto de vista del usuario, en el que se explica cuál es su finalidad, el tipo de usuarios que podrán utilizarlo y cómo puede usarse.

La arquitectura se explica en detalle en el Capítulo 5 así como la estructura de ficheros y el entorno web del proyecto. También se introducen los módulos principales de tratamiento de datos y de análisis de los mismos, que se tratarán en profundidad en capítulos posteriores.

En el Capítulo 6 se cuenta todo lo relacionado con la gestión y el almacenaje de los datos aportados por el usuario. Esto incluye también una pequeña base de datos de la que también se hablará.

Por último, relativo a implementación, está el Capítulo 7, en el que se trata la visualización de estadísticas, el sistema predictivo y la agrupación de los datos.

Las conclusiones finales del proyecto, así como los trabajos a realizar en el futuro para completar el sistema se incluirán en el Capítulo 8.

El Capítulo 10 contiene la descripción de las aportaciones al proyecto de cada uno de los integrantes.

Capítulo 2: Introduction

Capítulo 2. Introduction

2.1 Motivation

The present project allows the teacher to track their students evolution, search for groups of students with similar behaviour and predict a final grade for each of them, based on data from previous years. Our main goal is helping the teachers and make the most of the enormous quantity of data from previous years.

Nowadays there are different kinds of projects using statistical techniques or techniques based on machine learning to give their users a global view of the systmen and predict situations or data that help them in the decision-making process. For example, the analysis applications of the financial systems show statistica graphs and predict economic results. A management team would be able to make better decisions with the help of these data. This tools make a grand impact and are now a fundamental part of the system.

However, these techniques can be useful in more sectors, not only the Economy. They are use in other kind of projects. For example: Search engines, antivirus, text comprehension apps, image exploration apps, etc.

Despite how well these techniques have worked in these other sectors, there are few lines of work using them focused in the academic enviroment to date.

If we look at the social contex of the academic enviroment, we can see that there have been many studies investigating the causes and factors that take students to give up their respective degrees. Based on these studies, we can highlight the following factors: Socio-economic problems, ignorance of the degree before entering it or even lack o the adequate previus knowledge.

Nevertheless, one of the factors with the highest recurrence rate we can find is the lack of an adequate following-up during the school year. The main reasons for this lack of following-up are:

- Ignorance of the rhythm or particularities of a specific subject has the consequence o poor time management.

- The level of difficulty for a teacher to track their students performance during the year.

All these causes and factors and the social and technological context come to the conclusion that using a statistical / predictive system applied to the academic follow-up would improve academic performance on both sides.

The academic results of a student are related to a series of socioeconomic variables and a pattern of behavior. The previously mentioned variables are usually unknown to the teacher, but they know performance of the students in their classroom. The attitude of a student to the subject can be predicted based on previous experiences, to the point of getting to know, roughly, what the final qualifications of a student could be.

As we will see below, there is a great interest in bringing these analysis methodologies to the academic field.

2.1.1 Access to university

There haver been several studies looking for a way of predicting academic success to modify the criteria for University access in Spain over the years. The objective is replace of devaluate the University Acess Test (PAU in Spanish), as this test doesn't really show weather or not the student is going to perform correctly during their time of Higher Education. We can find this idea, for example, in "La predicción del rendimiento como criterio para el ingreso en la universidad" [1]. This work recommend using the marks of the last three years of Secondary Education according to the academic center of origin and the desired one to predict, through multiple regression, whether the student will make the most of their time.

The Universities opened in Chile before 1981 are using the results of a University Selection Test (PSU in Spanish) weighted with the average of the Marks of Seconday Education (NEM in Spanish) since 2003. With this info they generate a ranking and predict the future academic performance of the student, to decide if said student can enter the University they are interested in. As shown in the study "Predicción de notas en Derecho de la Universidad de Chile: ¿Sirve el ranking?" [2], the predictive systems works as expected. However, it takes into account the socioeconomic variables more than it initally seems.

University entrance exams are carried out every year in India. These exams determine if a student can enter a certain University center. These tests are a determinant factor in the quality of the education the students will receive in the future. Due to the delay in the results, there are many predictors online that allow the student to get an idea of their near future. They usually ask for the date of the exam and the expected grade. Some of them ask for a test code as well. Unlike in Chile, these predictions ara not a condition for going to College [3] [4].

2.1.2 In the classroom

The study "La predicción del rendimiento académico: regresión lineal versus regresión logística" [5] briefly discusses previous works and compare multiple linear regression and logical regression. Using grades, class attendance, participation in the class and a survey with different questions about the situation of the student before starting their university

education. The author concludes that the best type of predictive technique is logical regression.

There is also Front Row Education [6], a tool that allows the teacher to analyze their students progress on the subject and give them extra material to work. The teacher has all the information they need in table and graph format, both by group and by student. Moreover, the teacher can change the language for the extra material an modify the standard of evaluation.

2.2 - Goals:

The project presented in this document aims to create a prototype for a tool that allows the teacher to track their class. This track will be done through statistical visualization, grouping of students with similar behaviors and predicting of grades. Although the development will focus on a specific subject (Development of Interactive Systems (DIS)) the system will consist of a modular architecture that allows ofr easy expansion. This way, it can be adapted to new subjects and new functionalities can be added.

The project will be based in the visualization of the data in a user friendly format, the creation of behavioral models of the students taking a given subject and the use os machine learning techniques to try and predict the final mark of a student.

In this version of the project we focus on the development of the internal architecture of the system an the functionality for the teacher, but with the forethought to make easy that, in future works, we can give functionality for the students.

Our goals for the implementation of the system are:

- The tool must be able to obtain a a series of data from a number of excel or csv files in a previously defined format and transform them in information that is easy to use and relevant for the system.

- The system should include tools for the display of statistical graphs from the data the teacher provides about the course.

- The system should be able to detect groups of students with similar behaviors and show a visualization of said groups through graphs. In addition, a list will be displayed indicating which group each student is in.

- A simple prediction tool will be provided. This tool should show, for each student in the class, a prediction of their a prediction of their possible mark, based on their performance in the subject and comparing it to the performance of students of previous years. This mark would be among one of the following: A, B, C, F.

The objective of the last point is investigate the feasability of the predictive algorithms in this enviroment. Although the project will include a simple prediction system, this part of the project should be studied and improved in future works.

With these goals fulfilled, the system should offer a tool to monitor and improve the performance of the students in a given course.

2.3 - Structure of the Report:

Chapter 3 explains the way we divided the time for the projec in stages and what jobs we did in any one of them. Besides, we make a brief review of all the tools and libraries used for the development of the work.

Chapter 4 describes our system from the point of view of the user, explaining its purpose, the kind of users it is aimed for and how it can be used.

The architecture, the file structure and the web enviroment of the project are explained in detail in Chapter 5. This chapter also introduces the main modules of data processing and analysis. These modules will be discussed in depth in later chapters.

Chapter 6 tells us everything related to the management and storage of the data provided by the user, including a small database.

Chapter 7 deals with the implementation: the system vstatistical visualization, for predicting and grouping data.

The final conclussions of the project, as well as future work to complete the system, will be included in Chapter 8.

Chapter 10 contains the description of the contributions to the project of each one of the team members.

Capítulo 3: Metodología y tecnologías empleadas

Capítulo 3: Metodología y tecnologías empleadas

En este capítulo se presentará la metodología seguida por los integrantes del grupo para el desarrollo del proyecto. Además, se hablará brevemente de las tecnologías o herramientas que se han utilizado en la implementación.

3.1 - Gestión del proyecto

El presente proyecto se difunde bajo licencia GNU GPL versión 3. Así mismo, el contenido multimedia generado por los autores, se difunde bajo licencia Creative Commons BY SA versión 3.0 (CC BY-SA 3.0).

A continuación, vamos a presentar la metodología seguida en la gestión de este proyecto, así como las herramientas utilizadas para tal efecto. Como objetivo de este capítulo, se pretende dotar al lector de una visión global del seguimiento llevado a cabo en el desarrollo de la aplicación.

Para la gestión interna del proyecto, en lo que a comunicación entre integrantes se refiere, se ha utilizado la herramienta Trello [7], un gestor completo de tareas que permite realizar un seguimiento de las mismas. Esta herramienta nos permite agregar comentarios y sincronizarse automáticamente con cualquier dispositivo en el que se tenga instalado de forma sencilla.

En cuanto al almacenamiento de la información relativa al proyecto, se ha realizado en su totalidad a través del servicio Google Drive, plataforma de almacenamiento en la nube. Además, en lo que respecta al código, se ha seguido un estricto sistema de versiones.

Dicho sistema de versiones consta de cuatro cifras separadas por '.' del estilo 'a.b.c.d'. El significado de cada cifra refleja el valor de importancia de la modificación realizada en la versión que corresponda. Los valores situados a la izquierda son los más relevantes y decrementando en importancia hacia la derecha.

Como entornos de desarrollo se han utilizado las siguientes herramientas:

- SublimeText (como editor de texto)
- MySQLWorkbench y XAMPP (para la gestión de la base de datos)
- Herramientas de depuración de Chrome y Firefox (como soporte al diseño web)
- MyBalsamiq y Lucidchart para realizar diagramas y esquemas de diseño
- Ventana de Comandos (como vía para lanzar o parar el servidor, así como depurar el sistema)

Cabe destacar que en todas las fases de pruebas se ha seguido una metodología cruzada. De esta forma, una vez implementada una funcionalidad concreta, la prueba en profundidad la realiza otro integrante del grupo. Con esto conseguimos una visión más global del sistema y no limitar la prueba al punto de vista del que desarrolló dicha funcionalidad.

Para continuar, pasamos a explicar las fases o iteraciones en las que se fundamenta todo el proceso del desarrollo de la aplicación. En dicho proceso podemos distinguir tres iteraciones principales. Los objetivos se fueron definiendo conforme a las decisiones que se iban tomando según avanzaba el desarrollo. De este modo, pasamos a explicar dichas iteraciones:

3.1.1 - Fases del proyecto

Iteración 1

Objetivos:

- Elección entre los lenguajes Python y R para análisis de datos
- Aprender cómo utilizar las herramientas que van a usarse
- Implementación de un cargador
- Elección de las herramientas empleadas para la arquitectura web
- Investigar los algoritmos de Clustering necesarios para crear el clusterizador

Periodo: (Octubre - Enero)

Desarrollo:

En la primera fase del proyecto, se definió una primera etapa de estudio e investigación sobre las herramientas a utilizar en el proyecto. Además de una fase de pruebas en diferentes tecnologías para comprobar cuáles de dichas tecnologías se ajustaban mejor a las necesidades del proyecto.

Referente a esto, se analizó qué lenguaje de programación se ajustaba mejor a los fuertes requisitos de análisis en los que destaca el proyecto. Se acotó la lista de posibles elecciones a Python y R. No obstante, tras una extensa comparativa se decidió el uso de Python por varios motivos, aunque destacan entre todos ellos, su facilidad de integración en entornos web y la calidad de sus librerías de análisis.

Tras dicha elección, se procedió a aprender el lenguaje y todas las plataformas auxiliares utilizadas, tales como sus librerías de análisis o el microframework utilizado para su integración.

Posteriormente, se analizaron los datos de entrada de los que disponíamos para el desarrollo de la aplicación. Con estos datos, implementamos un cargador local que tratase toda la información. Con esto, ajustamos los datos en una serie de ficheros con la información relevante debidamente ordenada y tratada.

Para analizar los datos, se realizaron estadísticas y visualización de gráficas con la librería matplotlib de Python. Dichas gráficas posteriormente quedaron en desuso ya que se decidió utilizar la librería Highcharts para mostrarlas a través del entorno web.

Por último, en esta iteración, se empezó la investigación sobre clustering. Se buscó información sobre los distintos algoritmos que existen y las implementaciones disponibles de cada uno de ellos.

Iteración 2

Objetivos:

- Diseño e implementación del entorno web
- Implementar y probar la funcionalidad que obtenga datos estadísticos como medias o desviaciones típicas.
- Integrar Highcharts para la visualización de las estadísticas.
- Profundización en los conceptos de Clustering
- Implementación de un prototipo funcional de Clustering.
- Estructuración de la memoria y aporte de información a la misma.
- Investigación de los algoritmos utilizados para predicción.

Periodo: (Enero - Abril)

Desarrollo:

Tras haber cumplido los objetivos de la primera iteración, comenzamos con el diseño e implementación del apartado visual de la aplicación web. Nos centramos en la parte referente a la presentación de estadísticas, generadas a partir de los datos de los alumnos. Posteriormente, se le muestran al profesor que imparte la asignatura. Tales estadísticas, como veremos en capítulos sucesivos de este documento, se dividen en Notas, Asistencias y Accesos al campus virtual.

Una vez implementado el aspecto visual de dichos apartados, se pasó a portar todo ese código de generación de estadísticas, realizados en la iteración anterior, a la estructura propia de la aplicación web. Para ello, requerimos del uso de la librería Highcharts, que configuramos e integramos debidamente en la aplicación.

Con este apartado visual cubierto, comenzamos con la implementación de los algoritmos de clustering. De esta forma, empezamos a formar grupos de alumnos que tuvieran conductas similares. Para la comparación de conductas, se utilizaron los datos obtenidos del proceso de carga. Estos datos son columnas de ficheros excel tales como notas en determinados ejercicios, asistencias en días concretos o número de accesos al campus virtual.

Tras realizar una primera versión funcional del clusterizador y agregar su correspondiente visualización gráfica a través de Highcharts, comenzamos a estructurar la memoria. Seguimos agregando contenido relevante generando los primeros borradores simples de la misma.

Como parte final de la iteración se dedicó tiempo al estudio y comprensión de los algoritmos de predicción. Dichos algoritmos se requieren para cumplimentar los requisitos del proyecto. Después, finalizamos con la generación de casos de prueba para el código generado hasta el momento.

Iteración 3

Objetivos:

- Diseño e implementación una base de datos utilizada para la gestión de permisos
- Gestión de roles y permisos en la aplicación web
- Implementación de los algoritmos de predicción
- Fase de pruebas en puntos críticos de la aplicación

Periodo: (Abril - Junio)

Desarrollo:

Para conseguir una versión navegable de la aplicación web, se ha desarrollado una base de datos que almacene la información referente a la gestión de usuarios. Además, dicha base de datos gestionará los tipos y rutas de los ficheros. De esta forma, se permite implementar una capa de seguridad que limite el acceso cuando sea necesario, así como asociar a cada profesor, únicamente, los ficheros de los alumnos a los que imparte clase.

Una vez implementada la base de datos correspondiente, se ha diseñado, tanto la capa de acceso a la misma, como toda la capa de gestión de usuarios.

Se agregó la funcionalidad de predicción de comportamiento. Esta funcionalidad intenta, a raíz de un conjunto de datos incompletos, predecir la nota que obtendrá el alumno al final del curso lectivo. La nota se da en unos valores acotados predefinidos (Suspenso, Aprobado, Notable, Sobresaliente). Aunque ha producido buenos resultados en las fases de pruebas, queda con margen de mejora e investigación para trabajos posteriores.

Tras tener una primera versión funcional de todo el proyecto, comenzamos la fase de pruebas y estudio de los resultados obtenidos.

Para finalizar, cabe destacar que la mayor parte de la memoria ha sido realizada en esta iteración.

3.2 Herramientas

- **Python**

En este apartado hablaremos del lenguaje de programación que se ha utilizado como base para este proyecto. Con esto, se pretende plasmar los motivos de dicha elección.

Python es un lenguaje de programación de propósito general que se fundamenta en la sencillez de su sintaxis. El objetivo es obtener aplicaciones cuyo código fuente sea fácilmente legible y, como consecuencia, de fácil mantenimiento.

De este modo, a día de hoy existe toda una filosofía de sencillez y "belleza" de código que rodea al lenguaje. Esto último puede verse en "*El Zen de Python"* por Tim Peters [8] (desarrollador de Python) en el cual define los principios en los que se fundamenta dicho lenguaje.

Además de esto, Python es extraordinariamente versátil en el tratamiento de datos y cálculos complejos. Además, posee otras características de peso tales como las que notamos a continuación:

- Es un lenguaje multiplataforma por lo que su capacidad de portabilidad de código es alta.

- A pesar de ser un lenguaje no compilado, su implementación de la gestión de memoria, basada en conteo de referencias, hacen de Python un lenguaje con un grado alto de eficiencia.

- Posee una de las comunidades más activas actualmente en el desarrollo y avance del lenguaje

No obstante, una de las cualidades más importantes de Python es el importante número de módulos y librerías que existen y que extienden su funcionalidad. Con respecto a esto, el

soporte de instalación de las librerías y módulos se realiza, en la mayoría de las ocasiones, de forma sencilla a través de la herramienta "pip" proporcionada para tal efecto.

Como se puede ver en "Python for Data Analysis" [9], entre algunas de las librerías más importantes, se encuentran *numpy (*contiene funciones para cálculos de carácter científico), *Scipy* (contiene funciones para cálculos numéricos), *OS (*proporciona portabilidad a la funcionalidad propia del sistema operativo) o *Pandas (*proporciona estructuras y análisis de datos de alto rendimiento).

Se ha decidido el uso de este lenguaje porque presenta gran facilidad para ser usado en entornos web y por la calidad de sus librerías de análisis de datos.

- **Librerías de Python**

A continuación, se va a dedicar un pequeño apartado a la correcta presentación de algunas de las librerías utilizadas en el proyecto debido a su gran importancia.

Las librerías de las cuáles se va a tratar en este apartado son "Pandas", dedicada al análisis de datos y "scikit-learn", dedicada a la minería y el análisis de datos, como se verá a continuación.

- **Pandas**

Pandas es una librería de software libre implementada para el lenguaje Python que se dedica al análisis de datos. Inicialmente fue desarrollada con el objetivo de gestionar datos financieros. No obstante, el desarrollo de la misma ha ido escalando con el paso del tiempo, dejándonos multitud de funciones útiles no sólo en dicho sector.

Esta librería posee una licencia de software libre "BSD" e implementa multitud de operaciones que facilitan el tratamiento de los datos. En la actualidad, es una de las extensiones al lenguaje más utilizada en multitud de proyectos que requieren de procesamiento eficiente de la información.

Este paquete contiene una serie de características que hacen de él, uno de los paquetes más usados en análisis de datos. Algunas de estas características son:

- Incluye estructuras de datos nuevas que facilitan el tratamiento de grandes cantidades de datos. Las estructuras de datos que incluye más importantes para nuestro proyecto son los DataFrame y las Series.

- Versiones más eficientes de estructuras ya implementadas para el almacenamiento de la información.

- Indexado multinivel que permite tener un potente control sobre agrupamientos de tablas y grandes colecciones de datos.

- Soporte sencillo para operaciones de entrada/salida a partir ficheros tales como Excel o ".csv"

- Generador de rangos de secuencias avanzado que brinda un amplio abanico de posibilidades para bucles con un mayor grado de control.

Gracias a las características que acabamos de mencionar, esta librería se adapta a la perfección con el presente proyecto.

- **scikit-learn**

Al igual que Pandas, scikit-learn es una librería implementada para el lenguaje Python y que cuenta con una gran comunidad que soporta el desarrollo de la misma. Actualmente es la librería más usada en este lenguaje para proyectos que incluyan técnicas de aprendizaje automático.

Proporciona gran cantidad de funcionalidades en minería y análisis de datos estando, además, perfectamente estructuradas en sus campos de actuación. Estos campos, como podemos ver en su sitio web [10], son los siguientes:

- Clasificación
- Regresión
- Clustering
- Reducción de las dimensiones de los datos
- Selección y generación de modelos
- Pre-procesamiento

De este modo, esta librería nos permite implementar toda la parte del proyecto dedicada a la predicción de notas y al clustering.

- **HTML, CSS y JavaScript**

Este conjunto de lenguajes, unidos entre sí, son frecuentemente usados para desarrollo de aplicaciones web. Para ello, además, es habitual utilizar PHP como lenguaje del lado servidor. De esta forma, es posible desarrollar sitios web complejos.

Sin embargo, en este proyecto se suprime el uso de PHP para reemplazarlo por Python como lenguaje del lado del servidor.

Por lo tanto, como iremos viendo en capítulos sucesivos de este documento, se utilizarán:

- HTML5 como lenguaje encargado de la organización estructural de la web.

- CSS como hoja de estilos que nos brinde un atractivo visual acorde a las necesidades.

- JavaScript como lenguaje del lado del cliente que dé soporte a la visualización (en su mayoría gráficas).

- Python como lenguaje del lado de servidor que realice todo el procesamiento de datos e implemente toda la lógica de la aplicación.

- **Highcharts**

Es una librería gráfica de código abierto implementada en JavaScript. En función del uso que se le quiera dar cuenta con distintos tipos de licencia que se ajustan a las necesidades de cada perfil solicitado[1]. Como podemos ver en dicho enlace, existe una licencia para perfiles no comerciales que se ajusta a las necesidades de este proyecto.

El objetivo de esta librería es facilitar la inclusión de gráficos interactivos en páginas web mediante el uso de una sintaxis sencilla. Cuenta con una gran cantidad de gráficos ya programados que son visualmente atractivos. Estos gráficos cuentan con una serie de parámetros que podemos modificar para adaptarlos a los datos que se quieren mostrar.

Es compatible con la mayor parte de los navegadores web modernos, tanto de escritorio como de dispositivos móviles. Además, es fácil de integrar con Flask y con Python. Esto nos lleva a elegir esta herramienta para visualizar los datos, tanto estadísticos como de agrupamiento y predicción.

Esta herramienta posee una buena documentación [11], muy bien estructurada y proporciona un gran soporte de uso.

- **Flask**

Es un microframework de software libre que permite la integración de Python para el desarrollo de plataformas web [12]. Es un sistema que se sirve de Jinja2 [13] para la inclusión de código Python en templates HTML. De esta forma, comunica la vista con la lógica de la aplicación de una forma eficaz.

Posee una estructura sencilla, ligera y fácil de manejar. Esto nos permite ganar rapidez en el tiempo de respuesta. Además, actualmente existen multitud de librerías desarrolladas para este microframework que complementan la funcionalidad del mismo.

- **Flask-WTF**

WTForms es una herramienta diseñada para la creación de formularios proporcionando una API con multitud de posibilidades. El proceso de desarrollo se hace de una forma eficiente y sencilla para el desarrollador, permitiéndo a este centrarse en el tratamiento de los datos y

[1] En el espacio https://shop.highsoft.com/highcharts se describen cada una de sus licencias.

en otros aspectos de la aplicación. Una de las características más importantes de esta herramienta es el uso de Jinja2 como lenguaje de inclusión de código en los templates, al igual que Flask. Esto nos brinda un grado de integración elevado.

Flask-WTF [14] es una librería diseñada específicamente para Flask que integra WTForms [15] con este microframework para solventar cualquier leve complicación que hubiera podido surgir con la integración de ambos sistemas.

De esta forma, Flask se complementa con soporte CSRF, subida de ficheros al servidor, formularios básicos o re-CAPTCHA.

- **myBalsamiq**

Herramienta de prototipado rápido [16] utilizada para la creación de mockups durante el diseño de la página web. Facilita el diseño colaborativo y proporciona una buena cantidad de elementos prediseñados para desarrollar bocetos completos de la aplicación. Permite, además, interconectar los distintos mockups dentro de un proyecto para comprobar, a groso modo, la usabilidad del sistema final.

- **MariaDB/MySQL**

MySQL [17] es un Sistema Gestor de Bases de Datos (SGBD) originalmente desarrollado con licencia de software libre por MySQLAB. Actualmente MySQL es propiedad de Oracle y se distribuye bajo dos licencias, GPL y comercial. Tras el traspaso de MySQL a Oracle, los desarrolladores originales, continuaron con el proyecto bajo el nombre de MariaDB [18]. Ambos proyectos poseen un nivel de compatibilidad casi total.

Se han usado ambos SGBD para asegurar dicha compatibilidad y permitir el despliegue en ambos sistemas cuando sea necesario. De esta forma nos aseguramos una portabilidad total entre ambos.

Capítulo 4: Descripción general del sistema

Capítulo 4: Descripción general del sistema

En este capítulo se contará la finalidad del sistema, cuáles son los tipos de usuarios que podrán utilizarlo, así como sus requisitos mínimos.

4.1 Finalidad del sistema

El objetivo final del proyecto consiste en desarrollar una herramienta útil e intuitiva que facilite el seguimiento de un grupo concreto de una asignatura. Está dedicada a profesores y alumnos. Para esto, contamos con tres modalidades o funciones. Estas nos reportan datos de interés con respecto a dicho seguimiento: estadísticas, clustering y predicción.

La parte estadística consta de una serie de gráficas que muestran un cálculo estadístico de algunas variables. Estas variables están agrupadas en tres grandes grupos: asistencias, notas y accesos al campus virtual.

Por otro lado, pasamos a hablar de la funcionalidad de clustering. En dicho apartado, se provee al profesor de dos tipos de clustering o agrupamiento. El primero de ellos, basado en algoritmos de clustering jerárquico y el siguiente basado en algoritmos de clustering no jerárquico. Cada uno de ellos, a su vez, permite al usuario, por medio de un sencillo desplegable, visualizar en qué grupo se encuentra un determinado alumno.

Para finalizar con este subapartado, la predicción permite al profesor aprovecharse de la experiencia adquirida de cursos anteriores para predecir qué nota sacará cada uno de los alumnos que tiene matriculados.

En esta primera fase del desarrollo del sistema, se ha decidido centrar los esfuerzos en una sola asignatura. No obstante, se ha diseñado con vista a que en el futuro se pueda generalizar. La asignatura elegida es Desarrollo de Sistemas Interactivos (DSI) debido a sus peculiaridades. Esta materia, en el curso 2015-2016, presenta una serie de características de evaluación únicas, que hacen que su análisis requiera especial atención. Por un lado, se realizan tres ejercicios de evaluación continua a lo largo del curso, cuya media ponderada cuenta un 10% de la nota final. Por otro lado, cuenta con dos modos de evaluación: por examen o por proyecto. Si el alumno decide ir por proyecto, éste cuenta el 70% de la calificación. Además, debe cumplir con dos entregas previas a la final, que van calificadas y cuentan cada una el 10%; y asistir como mínimo al 80% de las clases teóricas y prácticas. En caso de ir por examen no es necesario cumplir con asistencia, entregas o proyecto y cuenta el 90 % de la nota final.

4.2 Usuarios

El presente proyecto está estructurado para soportar tres tipos de usuarios: administrador, profesor y alumno. No obstante, el sistema actual se centra en la funcionalidad propia del profesor y proporciona funcionalidad básica para el administrador. De esta forma, se deja la funcionalidad del alumno como propuesta para trabajo futuro.

Debido a esto, pasamos a explicar los usuarios que se encuentran actualmente con funcionalidad suficiente como para ser tenidos en cuenta:

- Administrador: tiene control total del sistema. Es el único que puede dar de alta a otros usuarios, asignaturas nuevas en el sistema, grupos que se imparten para cada asignatura y los alumnos matriculados en cada grupo. Además de la asignación de los profesores a dichos grupos.

- Profesor: proporciona al sistema la información evaluable de la que dispone sobre sus alumnos siguiendo un formato un específico que se explicará con detalle más adelante. En función de dicha información, recibe las estadísticas, agrupamientos creados por algoritmos de clustering y predicciones que le permitirán hacer un seguimiento de su grupo. Cabe destacar un mismo profesor podrá tener distintos grupos asignados en un mismo momento siendo capaz, en todo momento, de pasar de uno a otro con facilidad. No puede darse de alta en el sistema por sí mismo.

- Alumno: aunque esta funcionalidad no se encuentra desarrollada, explicamos el objetivo de la misma. Es capaz de ver las estadísticas del grupo y de clustering con el añadido de tener una pequeña marca que indique su posición en las mismas. Además de esto, podrá obtener una predicción simple de su nota final.

4.3 Flujo de actividad: común para todos los usuarios[2]

Lo primero que todos los usuarios encontrarán, estén registrados o no, es la pantalla de inicio.

[2] Todas las capturas de pantalla que se presentan en este apartado se ha realizado teniendo la colección de datos completa de un curso.

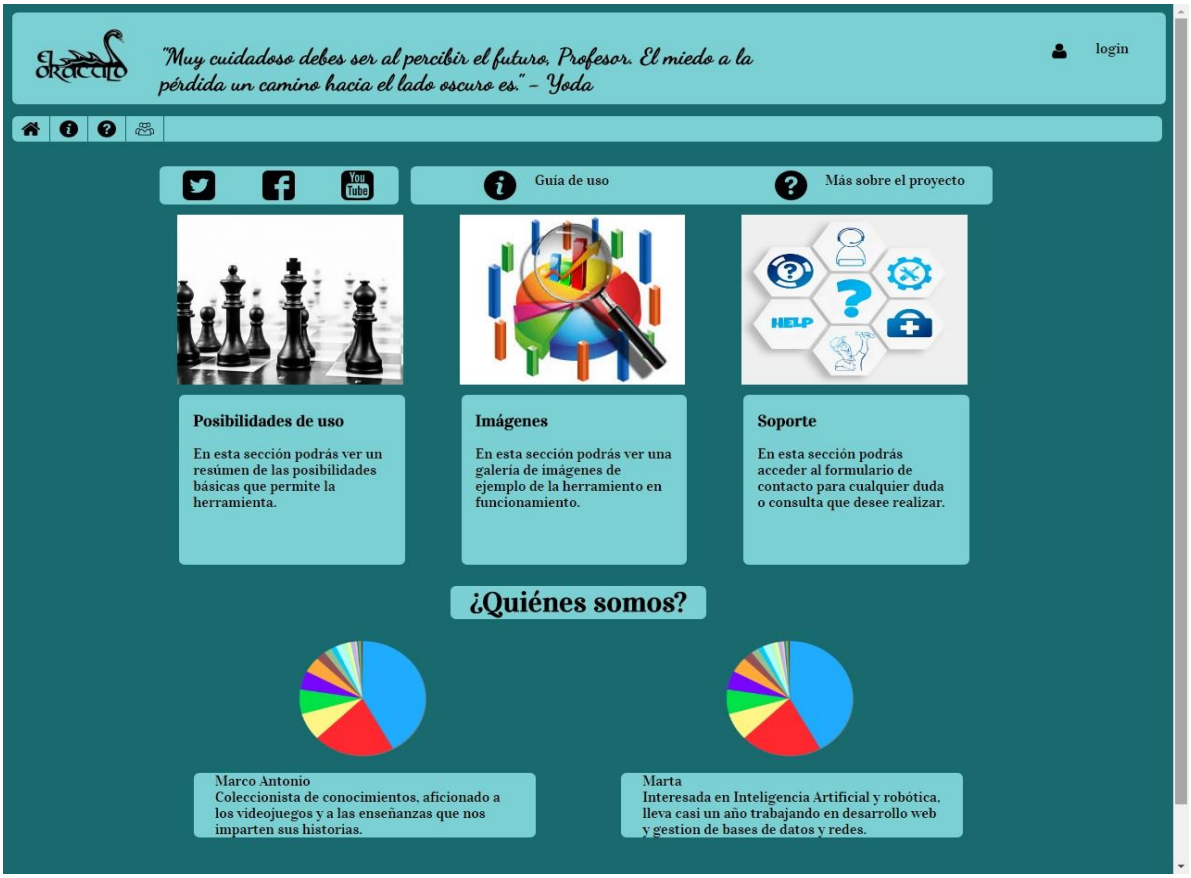


Ilustración 1: Página de inicio

En ella se puede encontrar un menú con opciones que llevan a otras pantallas: una con información sobre el uso del sistema, otra con las preguntas frecuentes y otra con información sobre los desarrolladores. Además, también aparece la opción de identificarse (login) en la esquina superior derecha.

Pulsando en dicha opción, la aplicación se dirige a la pantalla de identificación:

Ilustración 2: Página de login

Tras identificarse en el sistema, profesor y administrador pueden realizar una serie de trabajos o consultas específicas de su rol.

4.4 Flujo de actividad: Administrador

Bajo su nombre de usuario en la esquina superior derecha, encontrará un enlace con el nombre de “Panel de registros”.

Ilustración 3: Enlace a panel de registros

Este enlace le llevará a una pantalla con cuatro botones: “Registrar Usuarios”, “Registrar Asignaturas”, “Registrar Grupos” y “Alumnos-Grupos”. Cada uno de estos botones lleva a diferentes formularios que permiten gestionar el sistema.

Ilustración 4: Panel de registros

- Registrar Usuarios: esta opción permite registrar un nuevo usuario en el sistema. Se puede hacer de forma individual (Ilustración 5.2), rellenando un formulario, o de forma masiva a través de un archivo (Ilustración 5.3), tipo hoja de cálculo, con el nombre y el correo electrónico. La contraseña se genera automáticamente.

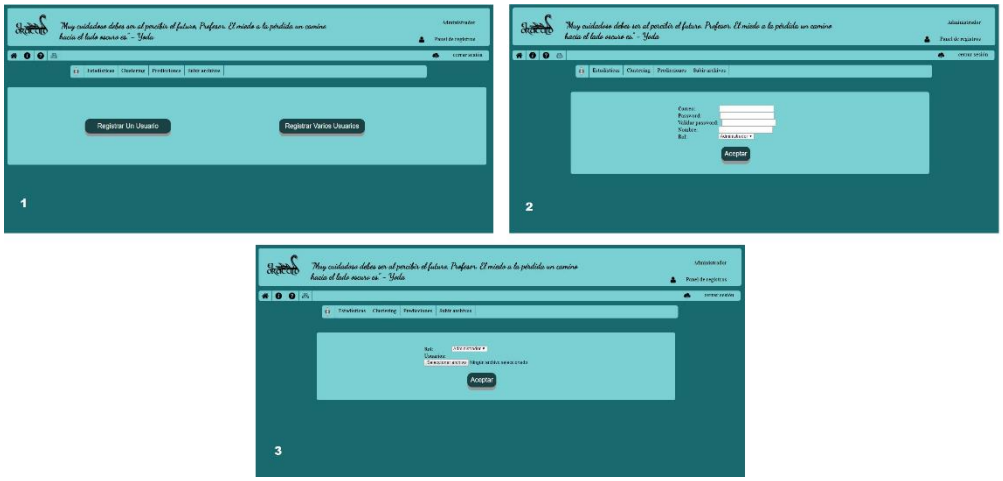


Ilustración 5: Registro de usuario

- Registrar Asignaturas: (Ilustración 6.4) registra una asignatura en el sistema. La información necesaria para hacerlo es el nombre y la descripción de la misma.

- Registrar Grupos: (Ilustración 6.5) esta opción permite registrar un grupo nuevo de una determinada asignatura. La información necesaria es el nombre del grupo, el año en el que se desarrolla, el profesor que la imparte y la asignatura a la que pertenece.

- Alumnos-Grupos: (Ilustración 6.6) esta opción permite registrar a qué grupo pertenece un alumno. Para hacerlo, hace falta indicar el grupo del que se desea hacer el registro y una lista de alumnos con sus nombres y sus correos electrónicos. En caso de que algún alumno que se quiera registrar en un grupo no esté dado de alta en la base de datos, se le incluye en el sistema automáticamente.

Ilustración 6: Registros

4.5 Flujo de actividad: Profesor

Lo más destacable para este tipo de usuario es un menú que encontrará en el centro de la pantalla. Lo primero de dicho menú es un botón con el nombre del grupo sobre el que se está trabajando en el momento y que, al pulsarlo, despliega un nuevo menú con la lista de grupos a los que imparte clase el profesor identificado. A continuación del botón de grupo, aparecen las siguientes opciones:

Ilustración 7: Menú del profesor

- Subir archivos: esta funcionalidad es la primera que debe emplear un profesor cuando empieza a trabajar con un grupo nuevo. Al elegirla, el sistema irá a una nueva pantalla con un cuadro de subida de ficheros y un botón de upload.

  Se espera que la información necesaria para trabajar la proporcione el profesor en forma de hojas de cálculo con un formato definido previamente. Los datos básicos que necesitará la aplicación para funcionar son:

  - ❏ Log de accesos al campus virtual
  - ❏ Las notas de las distintas prácticas que vayan teniendo lugar a lo largo del curso lectivo.

  Como documentación adicional, se pueden aportar las asistencias a clase y las notas de ejercicios realizados. Los ejercicios representan pequeños entregables evaluables mientras que las prácticas son tareas más grandes con mayor tiempo para su resolución y con una fecha de entrega fija.

Ilustración 8: Página de subida de fichero

- Estadísticas: esta funcionalidad permite al profesor analizar gráficamente la evolución de su grupo. Se le presenta la opción de ver las notas, los accesos y la asistencia sobre un eje de coordenadas, tanto de forma global como individualmente.

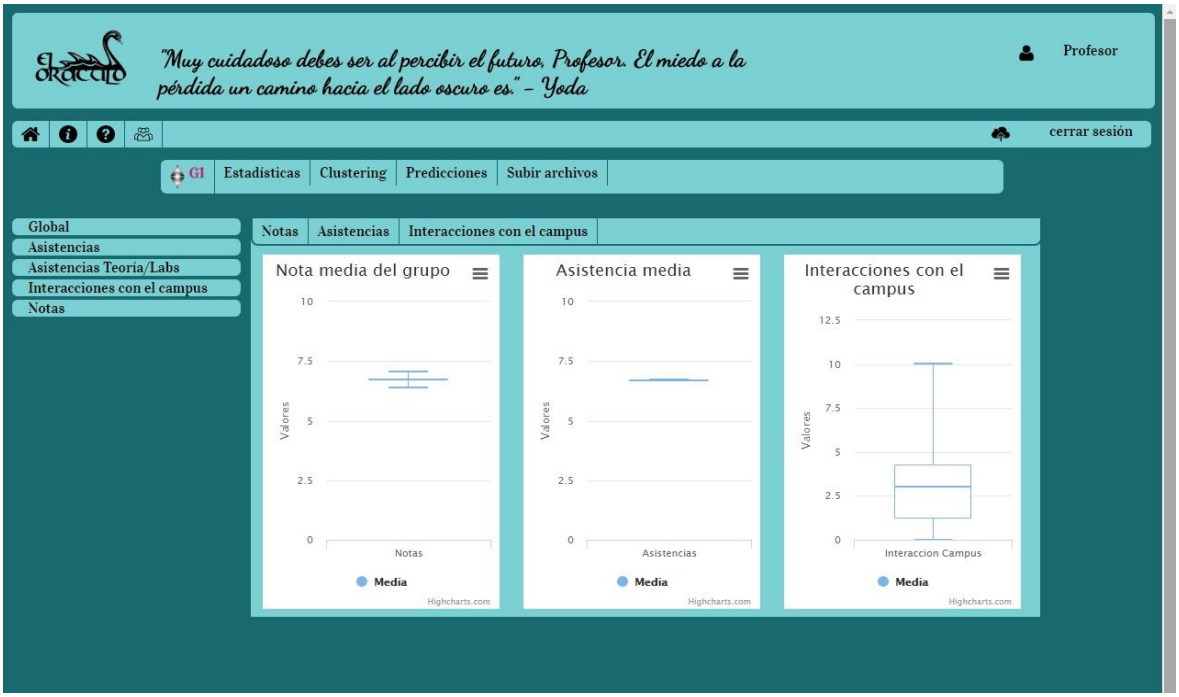


Ilustración 9: Página de gráficas generales

Además, en el menú lateral aparecen más opciones, como se muestran en la imagen que se encuentra bajo estas líneas. En la esquina superior izquierda está la gráfica que representa la asistencia sobre una línea de tiempo. A su derecha se muestra la misma información, pero diferenciando si la asistencia es a clase de teoría o de laboratorio. En la esquina inferior izquierda se presenta la información relativa a la media de accesos al campus virtual por día o por mes, según interese. Por último, en la esquina inferior derecha, aparecen las notas medias y sus variaciones de los distintos ejercicios y prácticas que se han ido realizando a lo largo del curso.

Ilustración 10: Gráficas de asistencias, accesos y notas

- Clustering: esta opción lleva a una pantalla intermedia que permite al profesor elegir qué tipo de clustering debe ejecutarse: jerárquico o no jerárquico. Cualquiera de los dos dividirá los alumnos en varios grupos según su comportamiento hacia la asignatura.

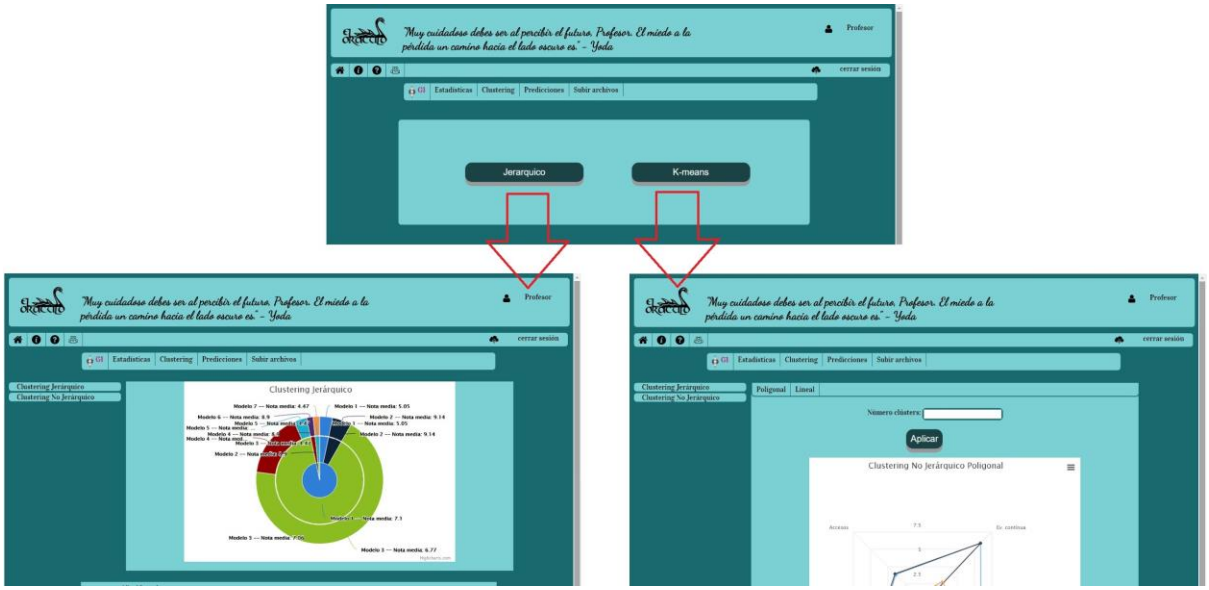
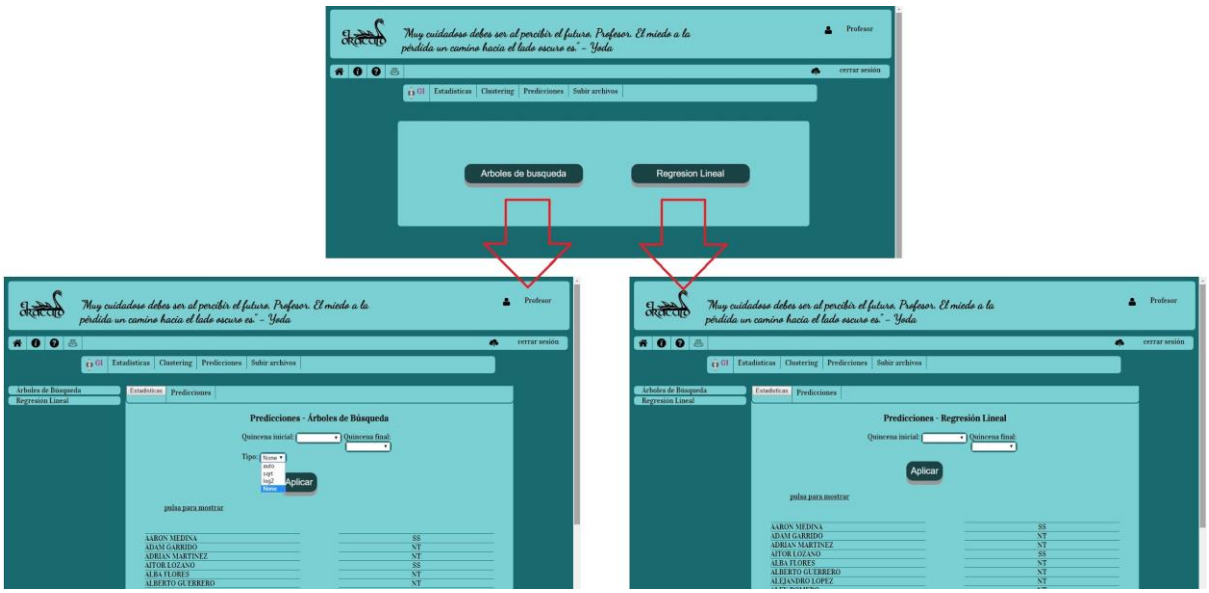


Ilustración 11: Páginas de clustering

- Predicción: esta opción lleva a una pantalla intermedia que permite al profesor elegir qué tipo de predicción desea que se lleve adelante, ya sea mediante árbol de decisión o mediante regresión lineal. El objetivo es poder extrapolar la nota final según el trabajo realizado hasta el momento, independientemente del método elegido.



[1]

Ilustración 12: Páginas de predicción

[1] Los nombres utilizados en las imágenes de este documento, no son reales.

Todas las funcionalidades del profesor, así como el tipo de datos y el contenido de los ficheros que debe subir, se tratarán más adelante en este mismo documento.

Capítulo 5: Arquitectura del sistema

Capítulo 5: Arquitectura del sistema

En el presente capítulo, se pasa a detallar la arquitectura general del sistema desarrollado para tal efecto.

Este proyecto posee un fuerte carácter de servicio, así como una necesidad de tener, en cada momento, datos actualizados sobre las asignaturas y entorno académico. Es por esto que se ha decidido realizar una aplicación web, como se verá más adelante. Además, esto nos permite conseguir un factor alto de accesibilidad para todos los usuarios que deseen usar el sistema.

5.1 Descripción general de la arquitectura

El presente proyecto posee una serie de módulos y partes diferenciadas que interactúan entre ellas para formar el sistema. En este espacio nos centramos en el aspecto funcional de la herramienta. Por ello, no entraremos en excesivos detalles tecnológicos de la misma.

Con esto, se pretende aclarar el funcionamiento interno de la aplicación a modo conceptual. Más adelante, dedicaremos un apartado explícito a la implementación de cada uno de ellos.

En la imagen se puede apreciar la arquitectura básica del proyecto. Se ha introducido un patrón visual para facilitar al lector la separación conceptual, en cuanto al apartado funcional, que posee el sistema. Para ello se ha usado un código de colores:

- El color azul consta de la funcionalidad relativa al tratamiento de datos y su correspondiente almacenamiento de forma permanente.

- El color verde consta de la lógica de la aplicación.

- El color naranja consta de la metodología empleada para conectar un proyecto web con el lenguaje Python.

- El color rosa consta de la vista y todos los elementos que interactúan para que se pueda producir en el formato deseado.

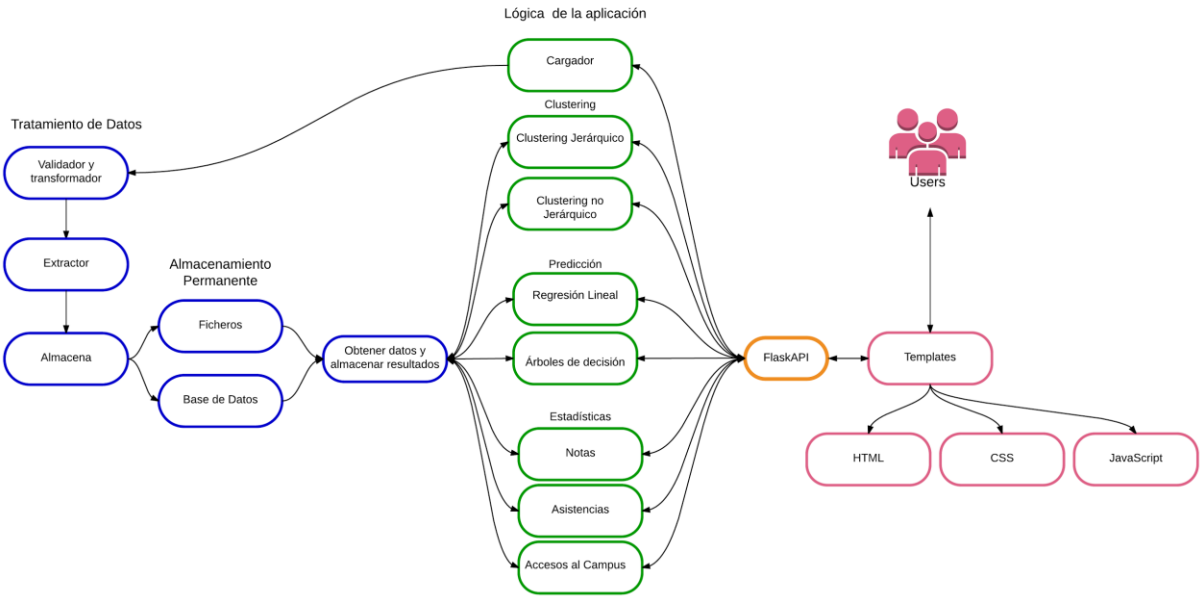


Ilustración 13: Arquitectura del sistema

Esta imagen muestra un modelo conceptual de la arquitectura de todo el sistema. Esta imagen es en la que se basará casi todo este capítulo.

El usuario solicita la página a la que desea acceder. El servidor (levantado por el propio microframework flask [12]) reconoce la URL introducida y la procesa. Una vez ha obtenido los datos, renderiza la página correspondiente mostrándosela al usuario.

Los datos se obtienen llamando al modelo de la aplicación que realiza funciones tales como obtener la estadística de asistencias de un grupo concreto o realizar la predicción de las notas de un grupo. Todo esto a raíz de los ficheros subidos previamente por el profesor o por el histórico de otros años.

Los ficheros de datos que se acaban de mencionar, se almacenan en el directorio ***"/datos".*** Son almacenados por el profesor que imparte el curso u obtenidas de otros cursos pertenecientes a la misma asignatura. Además de esto, se posee una base de datos que almacena los usuarios, así como sus roles, así como la vinculación de dichos usuarios a los ficheros de datos que se almacenan. El objetivo es hacer una gestión básica de permisos.

Una vez el usuario los selecciona para ser subidos a la aplicación, los ficheros pasan por un cargador que valida el correcto formato de dichos ficheros, así como que cumplan una serie de requisitos impuestos que se tratarán en capítulos posteriores. Una vez se ha completado el proceso de carga, pasa por un proceso de extracción que transforma los datos adecuadamente.

5.2 Tratamiento de datos

Este módulo representa todo el flujo de ejecución desde que el usuario selecciona los ficheros, que desea cargar al sistema, hasta que estos quedan convenientemente almacenados en sus respectivos subdirectorios. Si hacemos zoom en la imagen, podemos ver lo siguiente:

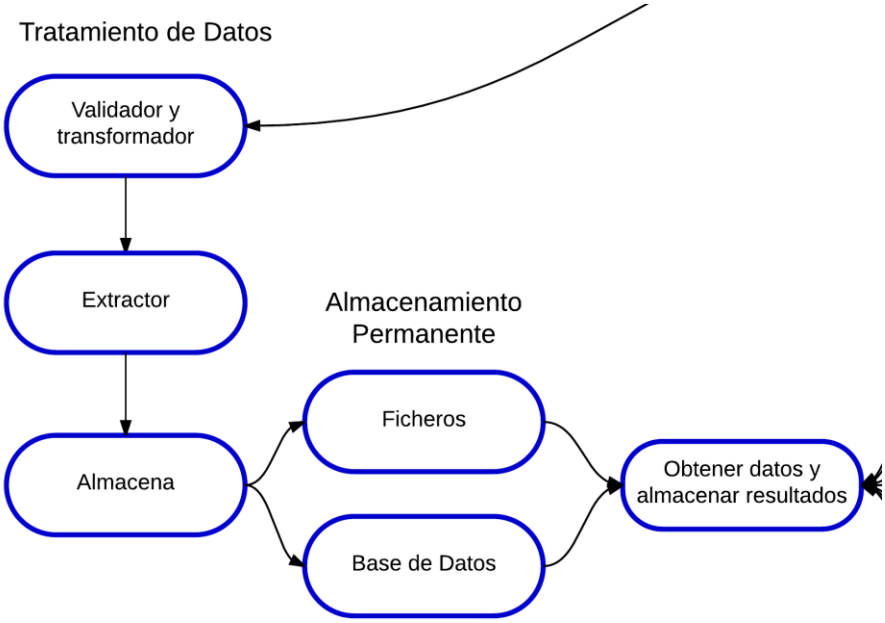


Ilustración 14: Tratamiento de los datos

Aquí podemos distinguir dos módulos principales que se encargan de la lógica de esta sección del programa:

- Validador transformador: Encargado de validaciones iniciales de los ficheros y adaptaciones leves de algunos ficheros.

- Extractor: Encargado de adaptaciones en profundidad de los datos para obtener nuevos datos de interés.

Todo este proceso será explicado en detalle en el capítulo 6.

5.3 Lógica de la aplicación

En esta clasificación englobamos los módulos o partes de análisis estadístico, clustering y predicción.

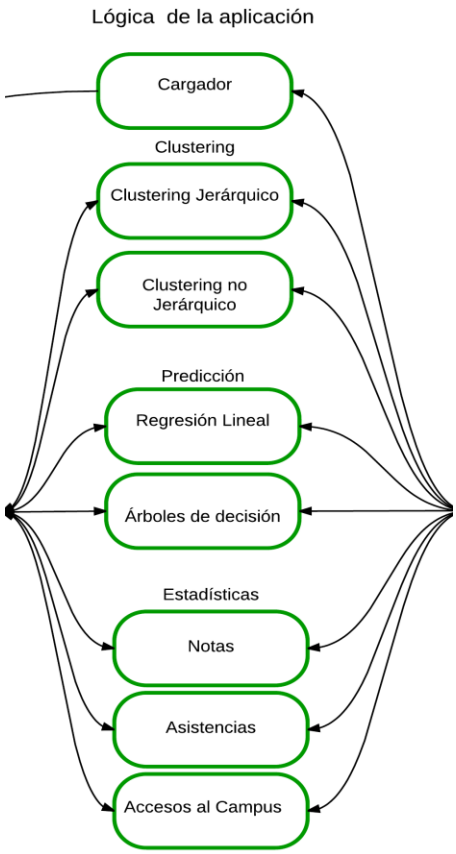


Ilustración 15: Lógica de la aplicación

El cargador incluido en esta parte de la lógica de la aplicación representa únicamente un conector con el proceso de carga que hemos vista en la sección anterior. Esto se realiza así para mantener un flujo estándar de la información. Por lo tanto, aunque en el diagrama de arquitectura incluimos el cargador en la lógica de la aplicación, no la tendremos en cuenta en este subapartado.

- **Submódulo de análisis estadístico**

- **Submódulo de análisis clustering**

- **Submódulo de análisis predictivo**

De toda esta sección se hablará con detalle en el capítulo 7.

5.4 Entorno Web

La aplicación será integrada en un entorno web que permita una interacción sencilla con el usuario. Los diseños iniciales de la interfaz web fueron realizados a través de la creación de mockups. Estos pueden verse en el apéndice D.

Aunque inicialmente se hicieron numerosas pruebas para integrar el sistema a través de otras tecnologías, finalmente se ha decidido utilizar un microframework llamado 'Flask' [12]. Este proporciona una gran sencillez de uso y funcionalidades adicionales además de una gran versatilidad.

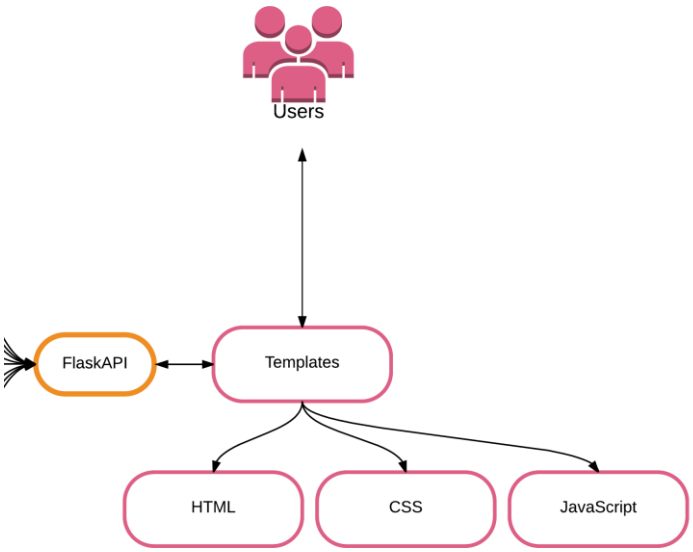


Ilustración 16: Estructura web

Se han diseñado un conjunto de páginas en las cuales no será necesario identificarse en el sistema para poder acceder, como puede ser el índice o la página de preguntas frecuentes.

También tendremos un conjunto de páginas en las cuáles será necesario identificarse con alguno de los roles descritos (Profesor o Administrador). Tras hacerlo, se les proporcionará la funcionalidad adecuada a su rol.

Para finalizar, se ha decidido el uso de la librería de javascript 'Highcharts' [11] que nos brinda una visualización elegante de los datos que deseamos mostrar. Además, posee como característica añadida, una integración sencilla con el framework utilizado.

Es importante destacar que el sistema web se servirá de "Flask" para ser renderizado. De esta forma, hará las veces de servidor que recibirá las peticiones y decidirá la página, con los parámetros concretos, que se debe renderizar.

Esto último nos permite tener el control total del sistema desde un único punto. Este punto será el servidor generado por "Flask". Así podremos derivar el entorno web a un plano

meramente de representación visual y comunicación de entrada/salida con el usuario del sistema.

Capítulo 6:
Tratamiento de datos

Capítulo 6: Tratamiento de datos

El presente capítulo explicará en detalle el flujo que siguen los datos desde que el profesor los introduce en el sistema hasta que finalmente son usados por la aplicación. Además de esto, se presentará el sistema de datos utilizado para la gestión de usuarios y permisos, así como la correlación entre usuarios y ficheros de datos.

Este capítulo se divide en varias secciones que intentan proporcionar al lector una visión precisa de cada parte referente al tratamiento de la información. A continuación, se expone una imagen que representa la interacción de los distintos componentes, así como una lista con los puntos que se tratarán a lo largo del capítulo que nos ocupa:

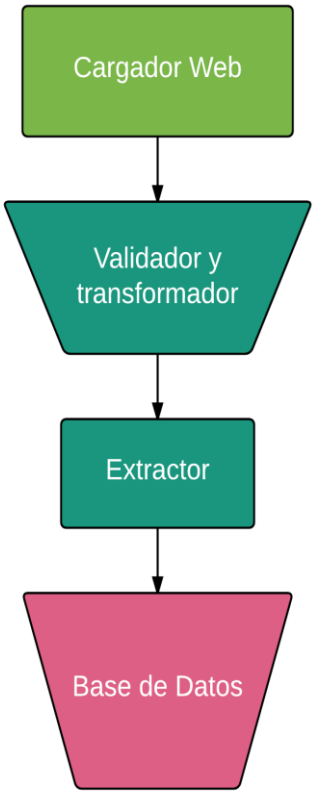


Ilustración 17: Proceso de carga de datos

- **Cargador web:** Parte encargada de la subida de los datos al servidor en el que se aloja la aplicación.

- **Validador y transformador:** Encargado del proceso de validación de los datos, así como de algunas transformaciones básicas.

- **Extractor:** Encargado del proceso de transformación de los datos.

- **Base de datos:** Desarrollada para la gestión de usuarios, permisos y relación entre ficheros y usuarios.

Como se puede ver, el flujo de procesamiento comienza con el cargador web, el cual tiene como objetivo la carga de los datos al sistema y, posteriormente, le pasa dicho flujo de ejecución al validador y transformador. Este, decide qué datos son válidos para el sistema y transforma debidamente la información. Posteriormente se le pasa el flujo al extractor, el cual agrega información al fichero de histórico de la asignatura. En caso de que dicho fichero aún no exista, lo crea. Para finalizar se almacena información referente a la gestión de los ficheros en la base de datos.

Por tanto, la transformación de los datos sucede en dos etapas: **Validador y transformador** y **Extractor**. Esto se realiza así para separar conceptualmente la transformación de la información, de la generación de nuevos ficheros con información calculada a raíz de los originales. A continuación, explicamos este mismo proceso con un poco más de detalle.

6.1 Proceso de carga

Para comenzar, es necesario que el profesor tenga obligatoriamente dos ficheros. El log del campus virtual **(accesos)** y el fichero de notas de prácticas y proyecto **(calificaciones)**. Los ficheros de ejercicios y asistencias son opcionales. Por lo tanto, para comenzar, el profesor debe descargarse el log del campus virtual de la siguiente forma:

Ilustración 18: Descarga del log 1

Primero seleccionamos la opción **“Registros”** que se encuentra en la sección **“Informes”** del panel **“Administración”**.

Después deberemos seleccionar el **“Log heredado”** para todos los participantes, todos los días, todas las actividades, todas las acciones y filtrando por el nivel de formación que se desee.

Ilustración 19: Descarga del log 2

Con esto, obtenemos el log del campus virtual **(accesos)** de todos los integrantes del curso. Cada fila de este fichero representa un click que el usuario ha realizado sobre algún enlace del curso. Como se puede ver en la siguiente imagen, tenemos una serie de datos para cada acceso. Son especialmente importante la hora (que posee no sólo la hora sino también la fecha en la que se produjo) y el nombre completo del usuario (usuario que realizó el click).

	A	B	C	D	E	F	G	H	I
1	Hora	Nombre completo del usuario	Usuario afectado	Contexto del evento	Componente	Nombre evento	Descripción	Origen	Dirección IP
2	30/06/2016 08:03	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
3	30/06/2016 08:03	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_unassign	role unassign Estudiante	web	0.0.0.0
4	15/12/2015 13:55	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
5	15/12/2015 13:55	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_unassign	role unassign Estudiante	web	0.0.0.0
6	30/11/2015 10:37	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
7	30/11/2015 10:37	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_unassign	role unassign Estudiante	web	0.0.0.0
8	30/11/2015 10:35	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
9	30/11/2015 10:35	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_unassign	role unassign Estudiante	web	0.0.0.0
10	30/11/2015 10:35	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
11	30/11/2015 10:35	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_unassign	role unassign Estudiante	web	0.0.0.0
12	7/10/2015 12:45	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
13	6/10/2015 08:31	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
14	4/10/2015 08:17	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
15	4/10/2015 00:49	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
16	1/10/2015 08:15	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
17	30/09/2015 14:05	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Profesor	web	0.0.0.0
18	30/09/2015 14:05	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_unassign	role unassign Estudiante	web	0.0.0.0
19	30/09/2015 14:05	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
20	30/09/2015 14:05	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Profesor	web	0.0.0.0
21	30/09/2015 14:05	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
22	30/09/2015 08:19	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0
23	30/09/2015 08:19	-	-	Curso: Desarrollo de Sistemas Interactivos - 2015/2016 - Grupo A	Sistema	role_assign	role assign Estudiante	web	0.0.0.0

Ilustración 20: Fichero de accesos (entrada)

Una vez que se ha obtenido este fichero, es necesario que el profesor haga un fichero excel con las notas de prácticas y proyecto **(calificaciones)** con la siguiente estructura:

	A	B	C	D	E	F	G	H	I	J	K
1	Nombre	Apellidos	Grupo de prácticas	P1	P2	Part.	Proy/Exa	Media	Nota	Cal	Observaciones
2				0,1	0,1	0,1	0,7				
3	SERGIO	ALONSO	G02	8,35	8,80	5,00	7,60	7,54	7,54	NT	
4	MARCOS	ALVAREZ	--	0,00	0,00	0,00	#N/A	#N/A	#N/A		
5	JUAN	BLANCO	P02	9,75	9,80	13,00	9,20	9,70	9,70	SB	
6	JULIA	CABRERA	P01	9,95	9,60	10,00	9,10	9,33	9,33	SB	
7	FRANCISCO	CALVO	--	0,00	0,00	0,00	#N/A	#N/A	#N/A		
8	GUILLERMO	CANO	--	0,00	0,00	0,00	#N/A	#N/A	#N/A		

Ilustración 21: Fichero de calificaciones (entrada)

Como se puede ver, este fichero contiene una lista con todas las notas que cada alumno ha obtenido en prácticas o proyectos, así como el grupo de proyecto al que pertenece. Posee además una columna "Nota" que es la nota final de este usuario. El fichero deberá ser subido con todas las columnas. En caso de que el curso no haya acabado, las columnas de las prácticas o calificaciones que no se hayan podido evaluar, deberán estar vacías.

Además de estos dos ficheros, tenemos otros dos ficheros opcionales que el profesor puede introducir. Ambos los deberá de crear el profesor con el formato que especificamos a continuación:

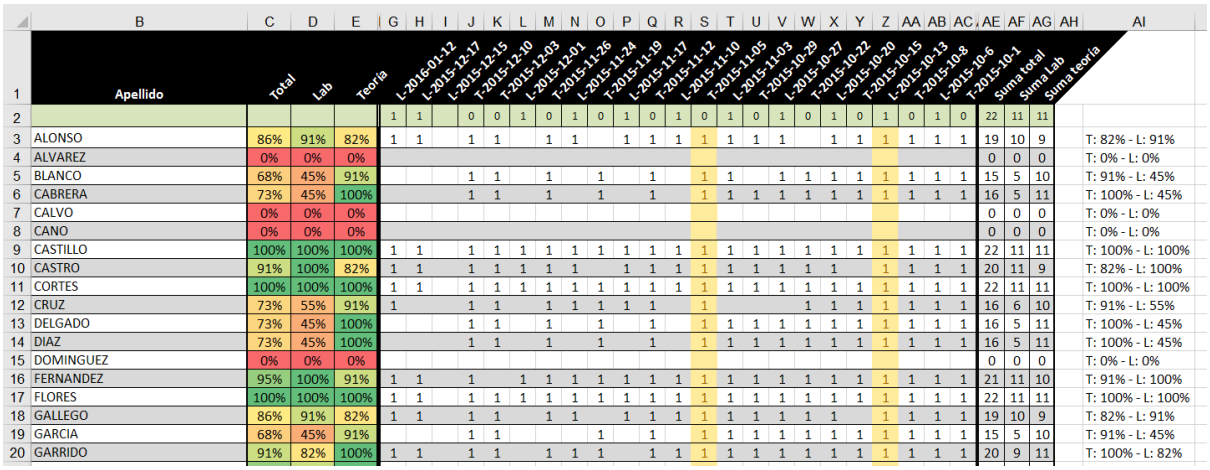
- ***Ejercicios***

	A	B	C	D	E	F	G	H	I
1	Nombre	Apellidos	Ej1 (0-3+1)	Ej2-OP (0-2)	Ej3 (0-4)	Asistencia > 80 > 90	Puntos		
2	SERGIO	ALONSO	3	0	0	2	5		82%
3	MARCOS	ALVAREZ		0	0	0	0		0%
4	JUAN	BLANCO	4	2	4	3	13		91%
5	JULIA	CABRERA	3	0	4	3	10		100%
6	FRANCISCO	CALVO		0	0	0	0		0%
7	GUILLERMO	CANO		0	0	0	0		0%
8	JESUS	CASTILLO	4	2	4	3	13		100%
9	BRUNO	CASTRO	4	0	4	2	10		82%
10	DYLAN	CORTES	4	2	4	3	13		100%
11	JOEL	CRUZ	4	0	0	3	7		91%
12	NICOLAS	DELGADO	3	0	4	3	10		100%
13	MATEO	DIAZ	3	1,5	4	3	11,5		100%
14	MIGUEL	DOMINGUEZ		0	0	0	0		0%
15	MARTIN	FERNANDEZ	3	1	3	3	10		91%
16	ALBA	FLORES	3	2	0	3	8		100%
17	LUCIA	GALLEGO	4	1,5	4	2	11,5		82%

Ilustración 22: Fichero de ejercicios (entrada)

Como se puede ver en la imagen, este fichero consta del nombre y los apellidos de los integrantes del grupo y una serie de notas obtenidas en los ejercicios evaluables.

- ***Asistencias***.



Apellido	Total	Lab	Teoría	L-2016-01-12	L-2015-12-17	L-2015-12-15	T-2015-12-10	L-2015-12-03	T-2015-12-01	L-2015-11-26	T-2015-11-24	L-2015-11-19	T-2015-11-17	L-2015-11-12	T-2015-11-10	L-2015-11-05	T-2015-11-03	L-2015-10-29	T-2015-10-27	L-2015-10-22	T-2015-10-20	L-2015-10-15	T-2015-10-13	L-2015-10-8	T-2015-10-6	T-2015-10-1	Suma total	Suma Lab	Suma teoría	AI
				1	1		0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	22	11	11	
ALONSO	86%	91%	82%	1	1		1	1		1	1		1	1	1	1	1	1	1		1	1	1	1	1	1	19	10	9	T: 82% - L: 91%
ALVAREZ	0%	0%	0%																							0	0	0	T: 0% - L: 0%
BLANCO	68%	45%	91%				1	1		1		1		1	1		1	1	1	1	1	1	1	1	1	1	15	5	10	T: 91% - L: 45%
CABRERA	73%	45%	100%				1	1		1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	16	5	11	T: 100% - L: 45%
CALVO	0%	0%	0%																							0	0	0	T: 0% - L: 0%
CANO	0%	0%	0%																							0	0	0	T: 0% - L: 0%
CASTILLO	100%	100%	100%	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	11	11	T: 100% - L: 100%
CASTRO	91%	100%	82%	1	1		1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1			20	11	9	T: 82% - L: 100%
CORTES	100%	100%	100%	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	11	11	T: 100% - L: 100%
CRUZ	73%	55%	91%	1			1	1		1	1	1	1	1	1				1	1	1	1	1	1	1	1	16	6	10	T: 91% - L: 55%
DELGADO	73%	45%	100%				1	1		1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	16	5	11	T: 100% - L: 45%
DIAZ	73%	45%	100%				1	1		1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	16	5	11	T: 100% - L: 45%
DOMINGUEZ	0%	0%	0%																							0	0	0	T: 0% - L: 0%
FERNANDEZ	95%	100%	91%	1	1		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	21	11	10	T: 91% - L: 100%
FLORES	100%	100%	100%	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	11	11	T: 100% - L: 100%
GALLEGO	86%	91%	82%	1	1		1	1		1	1		1	1	1	1	1	1	1	1	1		1	1	1	1	19	10	9	T: 82% - L: 91%
GARCIA	68%	45%	91%				1	1				1		1	1	1	1	1	1	1	1	1	1	1	1	1	15	5	10	T: 91% - L: 45%
GARRIDO	91%	82%	100%	1	1		1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	20	9	11	T: 100% - L: 82%

Ilustración 23: Fichero de asistencias (entrada)

Como se puede observar en la imagen, este último fichero consta de los nombres y apellidos de los integrantes del grupo. Cada columna que comienza por "L-" corresponde a un día de clase de laboratorio mientras que las columnas que comienzan por "T-" corresponden a un día de clase de teoría. Para cada alumno registramos un 1 si ha asistido al día concreto. El resto de columnas surgen como la suma o porcentaje de asistencias.

Una vez el profesor tiene los ficheros que quiere subir al sistema, procede a cargarlos al mismo a través del cargador web. Dicho cargador web es el formulario de subida de ficheros implementado a través de Flask. La implementación se detalla en el Apéndice C.

Después, esos datos pasan al validador y transformador. Aquí se comprueba que el fichero es válido (tanto en extensión como en formato) y se convertirá a un fichero '.csv' con el cuál se trabajará más tarde. Si no se cumple este requisito, la operación quedará abortada.

Tras haber comprobado dicho requisito, se comprueba el tipo de cada fichero introducido en función de cuatro tipos definidos **(*accesos, calificaciones, ejercicios y asistencias*)**. Para la correcta realización de esta fase, se comprueba que el fichero contenga las columnas concretas para cada uno.

Una vez que se ha identificado cada fichero con su correspondiente tipo, se comprueba que se hayan introducido los ficheros obligatorios para el correcto funcionamiento del sistema. Dichos ficheros corresponden a los tipos *accesos* y *calificaciones.* Para esto, se ha definido un sistema de almacenamiento por niveles que se explicará más adelante. De esta forma, los tipos de *ejercicios* y *asistencias* quedan a elección del profesor con carácter opcional.

En caso de que el fichero haya pasado todas las validaciones, se convertirá a formato CSV (si no se encuentran ya en ese formato). Se le dará un nombre adecuado a su contenido (*accesos, calificaciones, ejercicios, asistencias)*.

Tras realizar este proceso, los ficheros que se obtienen como resultado son los siguientes:

- Accesos

	Hora	Nombre completo del usuario	Usuario afectado	Contexto del evento	Componente	Nombre evento	Descripción	Origen	Dirección IP	Dia	Mes	Anho	Horario
2	30/06/2016 8:03	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	6	2016	8
3	30/06/2016 8:03	-	-	Curso: Desarrollo de Si	Sistema	role_unassign	role unassign E	web	0.0.0.0	30	6	2016	8
4	15/12/2015 13:55	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	15	12	2015	13
5	15/12/2015 13:55	-	-	Curso: Desarrollo de Si	Sistema	role_unassign	role unassign E	web	0.0.0.0	15	12	2015	13
6	30/11/2015 10:37	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	11	2015	10
7	30/11/2015 10:37	-	-	Curso: Desarrollo de Si	Sistema	role_unassign	role unassign E	web	0.0.0.0	30	11	2015	10
8	30/11/2015 10:35	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	11	2015	10
9	30/11/2015 10:35	-	-	Curso: Desarrollo de Si	Sistema	role_unassign	role unassign E	web	0.0.0.0	30	11	2015	10
10	30/11/2015 10:35	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	11	2015	10
11	30/11/2015 10:35	-	-	Curso: Desarrollo de Si	Sistema	role_unassign	role unassign E	web	0.0.0.0	30	11	2015	10
12	07/10/2015 12:45	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	7	10	2015	12
13	06/10/2015 8:31	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	6	10	2015	8
14	04/10/2015 8:17	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	4	10	2015	8
15	04/10/2015 0:49	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	4	10	2015	0
16	01/10/2015 8:15	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	1	10	2015	8
17	30/09/2015 14:05	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Prof	web	0.0.0.0	30	9	2015	14
18	30/09/2015 14:05	-	-	Curso: Desarrollo de Si	Sistema	role_unassign	role unassign E	web	0.0.0.0	30	9	2015	14
19	30/09/2015 14:05	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	9	2015	14
20	30/09/2015 14:05	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Prof	web	0.0.0.0	30	9	2015	14
21	30/09/2015 14:05	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	9	2015	14
22	30/09/2015 8:19	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	9	2015	8
23	30/09/2015 8:19	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	30	9	2015	8
24	29/09/2015 8:09	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	29	9	2015	8
25	29/09/2015 8:09	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	29	9	2015	8
26	26/09/2015 8:19	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	26	9	2015	8
27	25/09/2015 18:49	-	-	Curso: Desarrollo de Si	Sistema	role_assign	role assign Estu	web	0.0.0.0	25	9	2015	18

Ilustración 24: Fichero de accesos (salida)

- Asistencias

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	Nombre	Apellido	Total	Lab	Teoria	L-2016-01-12	L-2015-12-17	T-2015-12-10	T-2015-12-03	L-2	T-	L-	T-	L-	T-	L-	T-	L-	T-	L-	T-	L-	T-	L-	T-	L-	T-	Suma total	Suma Lab	Suma teoria
2	SERGIO	ALONSO	0.86363	0.9090	0.8181818	1.0	1.0	1.0	1.0		1.	1.0		1.	1.	1.	1.0	1.	1.	1.0		1.	1.	1.	1.	1.	1.	1.	19.0	10.0	9.0
3	MARCOS	ALVAREZ	0.0	0.0	0.0																							0.0	0.0	0.0
4	JUAN	BLANCO	0.68181	0.4545	0.9090909090909092			1.0	1.0		1.0	1.0	1.0	1.0	1.0	1.0	1.0			1.0	1.	1.	1.	1.	1.	1.	1.	1.	15.0	5.0	10.0
5	JULIA	CABRERA	0.72727	0.4545	1.0			1.0	1.0		1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	16.0	5.0	11.0
6	FRANCISCO	CALVO	0.0	0.0	0.0																							0.0	0.0	0.0
7	GUILLERMO	CANO	0.0	0.0	0.0																							0.0	0.0	0.0
8	JESUS	CASTILLO	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	22.0	11.0	11.0
9	BRUNO	CASTRO	0.90909	1.0	0.8181818	1.0	1.0	1.0	1.0	1.0	1.	1.0		1.	1.	1.	1.	1.	1.	1.	1.	1.0		1.	1.	1.	1.	1.	20.0	11.0	9.0
10	DYLAN	CORTES	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	22.0	11.0	11.0
11	JOEL	CRUZ	0.72727	0.5454	0.9090909	1.0		1.0	1.0		1.	1.	1.	1.	1.0	1.0			1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	16.0	6.0	10.0
12	NICOLAS	DELGADO	0.72727	0.4545	1.0			1.0	1.0		1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	16.0	5.0	11.0
13	MATEO	DIAZ	0.72727	0.4545	1.0			1.0	1.0		1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	16.0	5.0	11.0
14	MIGUEL	DOMINGUEZ	0.0	0.0	0.0																							0.0	0.0	0.0
15	MARTIN	FERNANDEZ	0.95454	1.0	0.9090909	1.0	1.0	1.0		1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	21.0	11.0	10.0
16	ALBA	FLORES	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	22.0	11.0	11.0
17	LUCIA	GALLEGO	0.86363	0.9090	0.8181818	1.0	1.0	1.0	1.0		1.	1.0		1.	1.	1.	1.	1.	1.	1.	1.	1.0		1.	1.	1.	1.	1.	19.0	10.0	9.0
18	HUGO	GARCIA	0.68181	0.4545	0.9090909090909092			1.0	1.0				1.0	1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	15.0	5.0	10.0
19	ADAM	GARRIDO	0.90909	0.8181	1.0	1.0	1.0	1.0	1.0		1.	1.	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	20.0	9.0	11.0
20	ANGEL	GIL	0.95454	0.9090	1.0	1.0		1.0	1.0	1.0	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	21.0	10.0	11.0
21	LUCAS	GOMEZ	0.86363	0.9090	0.8181818	1.0	1.0		1.0	1.0	1.	1.	1.	1.	1.	1.0		1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	19.0	10.0	9.0

Ilustración 25: Fichero de asistencias (salida)

- Ejercicios

	Nombre	Apellidos	Ej1	Ej2	Ej3	Media
0	SERGIO	ALONSO	3.0	0.0	0	3.333.333.333.333.330
1	MARCOS	ALVAREZ		0.0	0	
2	JUAN	BLANCO	4.0	2.0	4	11.111.111.111.111.100
3	JULIA	CABRERA	3.0	0.0	4	7.777.777.777.777.770
4	FRANCISCO	CALVO		0.0	0	
5	GUILLERMO	CANO		0.0	0	
6	JESUS	CASTILLO	4.0	2.0	4	11.111.111.111.111.100
7	BRUNO	CASTRO	4.0	0.0	4	8.888.888.888.888.890
8	DYLAN	CORTES	4.0	2.0	4	11.111.111.111.111.100
9	JOEL	CRUZ	4.0	0.0	0	44.444.444.444.444.400
10	NICOLAS	DELGADO	3.0	0.0	4	7.777.777.777.777.770
11	MATEO	DIAZ	3.0	1.5	4	9.444.444.444.444.440
12	MIGUEL	DOMINGUEZ		0.0	0	
13	MARTIN	FERNANDEZ	3.0	1.0	3	7.777.777.777.777.770
14	ALBA	FLORES	3.0	2.0	0	55.555.555.555.555.500
15	LUCIA	GALLEGO	4.0	1.5	4	10.555.555.555.555.500
16	HUGO	GARCIA	3.0	1.0	4	8.888.888.888.888.890
17	ADAM	GARRIDO	4.0	0.0	0	44.444.444.444.444.400
18	ANGEL	GIL	4.0	2.0	4	11.111.111.111.111.100
19	LUCAS	GOMEZ	4.0	1.5	4	10.555.555.555.555.500

Ilustración 26: Fichero de ejercicios (salida)

- Calificaciones

	Nombre	Apellidos	Grupo de prácticas	P1	P2	Part.	Proy/Exa	Media	Nota	Cal
2	SERGIO	ALONSO	G02	8.35	8.799.999.999.999.990	5.0	7.6	75.349.999.999.999.900	75.349.999.999.999.900	NT
3	MARCOS	ALVAREZ	--	0.0	0.0	0.0				NP
4	JUAN	BLANCO	P02	9.75	9.8	13.0	9.2	9.695	9.695	SB
5	JULIA	CABRERA	P01	9.95	9.6	10.0	9.1	9.325	9.325	SB
6	FRANCISCO	CALVO	--	0.0	0.0	0.0				NP
7	GUILLERMO	CANO	--	0.0	0.0	0.0				NP
8	JESUS	CASTILLO	G03	9.75	9.0	13.0	10.0	10.175	10.0	SB
9	BRUNO	CASTRO	G04	9.65	8.0	10.0	972.675	9.573.725	9.573.725	SB
10	DYLAN	CORTES	G01	8.65	7.200.000.000.000.000	13.0	8.643.347.916.666.660	8.935.343.541.666.660	8.935.343.541.666.660	NT
11	JOEL	CRUZ	G06	9.05	9.0	7.0	7.979	8.090.300.000.000.000	8.090.300.000.000.000	NT
12	NICOLAS	DELGADO	P01	9.95	9.6	10.0	7.25	8.03	8.03	NT
13	MATEO	DIAZ	P01	9.95	9.6	11.5	9.7	9.895	9.895	SB
14	MIGUEL	DOMINGUEZ	--	0.0	0.0	0.0				NP
15	MARTIN	FERNANDEZ	G04	9.65	8.0	10.0	952.875	9.435.125	9.435.125	SB
16	ALBA	FLORES	G03	9.75	9.0	8.0	8.478.750.000.000.000	8.610.125	8.610.125	NT
17	LUCIA	GALLEGO	G01	8.65	7.200.000.000.000.000	11.5	7.444.891.666.666.660	79.464.241.666.666.600	79.464.241.666.666.600	NT
18	HUGO	GARCIA	P02	9.75	9.8	11.0	9.0	9.355	9.355	SB
19	ADAM	GARRIDO	G06	9.05	9.0	7.0	76.629.999.999.999.900	7.869.099.999.999.990	7.869.099.999.999.990	NT
20	ANGEL	GIL	G03	9.75	9.0	13.0	99.999.375	1.017.495.625	10.0	SB
21	LUCAS	GOMEZ	G07	6.1	5.4	11.5	3.1	4.47	4.47	SS
22	DANIEL	GONZALEZ	G07	6.1	5.4	7.0	3.1	4.02	4.02	SS
23	ALBERTO	GUERRERO	G01	8.65	7.200.000.000.000.000	11.5	8.495.910.416.666.660	8.682.137.291.666.660	8.682.137.291.666.660	NT

Ilustración 27: Fichero de calificaciones (salida)

Con esto, ya tenemos los ficheros básicos que se utilizarán para generar las estadísticas. A continuación, generaremos un fichero nuevo a partir de estos.

Para esto, comenzaremos con la lectura de los ficheros que acabamos de ver. Posteriormente se adaptarán esos ficheros para que sean más fácilmente tratables. Se añadirán o eliminarán algunas columnas. Los cambios que se realizan a los ficheros, son los siguientes:

- Accesos: Se añaden las columnas “Día”, “Mes”, “Anho” y “Horario” extraídas de la columna “Hora”. Para ello, se separan los días, los meses, los años y las horas de la columna “Hora” y se generan las respectivas columnas.

- Asistencias: En este fichero eliminamos todas las filas o columnas que estén a null ya que no representan información útil. Además, generamos dos nuevos ficheros “asistencias-Teoria.csv” y “asistencias-Laboratorios.csv” separando las asistencias de teoría y de laboratorios respectivamente. No obstante, el fichero “asistencias.csv” se sigue manteniendo.

- Ejercicios: Puesto que tiene dos posibles formatos, se comienza obteniendo los valores de las columnas en su formato concreto. Posteriormente se modifican las columnas con un nombre estándar para ellas. Los nombres de las columnas son:

    - Apellidos
    - Ej1
    - Ej2
    - Ej3
    - Media

- Calificaciones: Puesto que tiene dos posibles formatos, se comienza obteniendo los valores para las columnas en las cuales se diferencian ambos formatos. Después se crea la columna correspondiente.

A continuación, comenzamos a crear el nuevo fichero que contiene los datos relevantes del resto de ficheros. Este nuevo fichero representa el histórico de la asignatura. Si ya está creado, simplemente se irá añadiendo información.

Para crear este fichero, primero se comprueba que los datos estén completos y se irán ajustando los nombres y quitando los acentos de los mismos.

Por último, los ficheros de *accesos*, *calificaciones*, *asistencias* y *ejercicios*, guardarán sus rutas en la base de datos vinculándose al profesor que ha subido los ficheros y a la asignatura a la que pertenecen. El fichero de *datos* guardará su ruta en la base de datos, vinculándose únicamente a la asignatura.

A continuación, mostramos tres capturas de pantalla para mostrar el fichero final de esta fase de carga. Se proporcionan tres imágenes debido a la excesiva longitud horizontal del fichero.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Nombre completo del usuario	Grupo de prácticas	P1	P2	Part.	Proy/Exa	Media	Nota	Cal	Quincena-1	Quincena-2	Quincena-3
2	AARON MEDINA	--	0.0	0.0	0.0				NP	0.0	0.0	0.0
3	ADAM GARRIDO	G06	9.05	9.0	7.0	76.629.999.999.999.900	7.869.099.999.999.990	7.869.099.999.999.990	NT	0.0	0.0	0.0
4	ADRIAN MARTINEZ	G02	8.35	8.799.999.999.999.990	11.5	7.6	8.184.999.999.999.990	8.184.999.999.999.990	NT	20.454.545.454.545.400	25.133.689.839.572.100	2.814.814.814.814.810
5	AITOR LOZANO	--	0.0	0.0	0.0				NP	0.0	0.0	0.0
6	ALBA FLORES	G03	9.75	9.0	8.0	8.478.750.000.000.000	8.610.125	8.610.125	NT	32.792.207.792.207.700	31.016.042.780.748.600	0.0
7	ALBERTO GUERRERO	G01	8.65	7.200.000.000.000.000	11.5	8.495.910.416.666.660	8.682.137.291.666.660	8.682.137.291.666.660	NT	0.0	0.8021390374331551	0.0
8	ALEJANDRO LOPEZ	G02	8.35	8.799.999.999.999.990	10.0	8.0	8.315	8.315	NT	17.857.142.857.142.800	5.240.641.711.229.940	2.444.444.444.444.440
9	ALEX ROMERO	P01	9.95	9.6	10.0	7.15	7.96	7.96	NT	0.19480519480519481	0.0	0.5185185185185185
10	ALVARO SANCHEZ	G05	9.4	8.8	9.0	83.655	857.585	857.585	NT	5.584.415.584.415.580	10.0	10.0
11	ANGEL GIL	G03	9.75	9.0	13.0	99.999.375	1.017.495.625	10.0	SB	12.337.662.337.662.300	28.342.245.989.304.800	0.0
12	ANTONIO RAMOS	G04	9.65	8.0	4.0	962.775	8.904.425	8.904.425	NT	6.136.363.636.363.630	4.545.454.545.454.540	4.888.888.888.888.880
13	BRUNO CASTRO	G04	9.65	8.0	10.0	972.675	9.573.725	9.573.725	SB	10.0	4.973.262.032.085.560	0.0
14	CARLOS NAVARRO	G02	8.35	8.799.999.999.999.990	9.0	8.0	8.215	8.215	NT	25.974.025.974.025.900	7.165.775.401.069.510	3.037.037.037.037.030
15	DANIEL GONZALEZ	G07	6.1	5.4	7.0	3.1	4.02	4.02	SS	0.6818181818181818	6.898.395.721.925.130	0.4444444444444445
16	DANIELA PENA	S01	9.4	8.000.000.000.000.000	3.0	5.0	5.54	5.54	AP	11.688.311.688.311.600	0.0	5.777.777.777.777.770
17	DARIO NUNEZ	P03	9.4	10.0	13.0	8.7	9.329.999.999.999.990	9.329.999.999.999.990	SB	275.974.025.974.026	0.6417112299465242	34.814.814.814.814.800
18	DAVID PEREZ	--	0.0	0.0	0.0				NP	0.0	0.0	0.0
19	DIEGO JIMENEZ	G07	6.1	5.4	8.0	8.850.000.000.000.000	8.145	8.145	NT	0.5194805194805195	5.454.545.454.545.450	2.740.740.740.740.740
20	DYLAN CORTES	G01	8.65	7.200.000.000.000.000	13.0	8.643.347.916.666.660	8.935.343.541.666.660	8.935.343.541.666.660	NT	33.766.233.766.233.700	3.048.128.342.245.980	16.296.296.296.296.200
21	ENZO VAZQUEZ	G07	6.1	5.4	4.0	4.65	4.805	4.805	SS	11.363.636.363.636.300	11.764.705.882.352.900	2.444.444.444.444.440

Ilustración 28: Fichero de datos 1

	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
	Quincena-23	Quincena-24	Numero Accesos	Suma total	Suma Lab	Suma teoría	Teo-Quincena-1-1	Lab-Quincena-1-1	Teo-Quincena-1-2	Lab-Quincena-1-2	Teo-Quincena-2-1	Lab-Quincena-2-1
	0.0	0.0	0.03000750187546887	20.0	9.0	11.0	0	1.0	0	0	0	0
	0.0	0.0	0.47261815453863465	21.0	10.0	11.0	0	1.0	0	0	0	0
48	4.381.625.441.696.110	5.069.767.441.860.460	3.968.492.123.030.750	20.0	11.0	9.0	0	1.0	0	0	0	0
	0.0	0.0	0.0075018754688672175	19.0	9.0	10.0	0	1.0	0	0	0	0
2.000	5.371.024.734.982.330	0.0	4.283.570.892.723.180	22.0	11.0	11.0	0	1.0	0	0	0	0
	34.275.618.374.558.300	874.418.604.651.163	37.284.321.080.270.000	0.0	0.0	0.0	0	0.0	0	0	0	0
6.200	31.802.120.141.342.700	26.976.744.186.046.500	39.984.996.249.062.200	0.0	0.0	0.0	0	0.0	0	0	0	0
	0.21201413427561838	30.697.674.418.604.600	12.303.075.768.942.200	15.0	5.0	10.0	0	0.0	0	0	0	0
1.100	9.116.607.773.851.590	544.186.046.511.628	8.192.048.012.003.000	22.0	11.0	11.0	0	1.0	0	0	0	0
8.100	0.0	24.651.162.790.697.600	18.154.538.634.658.600	16.0	6.0	10.0	0	1.0	0	0	0	0
3.580	657.243.816.254.417	1.069.767.441.860.460	6.031.507.876.969.240	15.0	5.0	10.0	0	0.0	0	0	0	0
	0.38869257950530034	22.325.581.395.348.800	39.234.808.702.175.500	19.0	10.0	9.0	0	1.0	0	0	0	0
1.130	35.689.045.936.395.700	19.534.883.720.930.200	4.021.005.251.312.820	0.0	0.0	0.0	0	0.0	0	0	0	0
	3.957.597.173.144.870	0.0	2.805.701.425.356.330	21.0	11.0	10.0	0	1.0	0	0	0	0
66	0.10600706713780919	0.6976744186046512	12.303.075.768.942.200	16.0	5.0	11.0	0	0.0	0	0	0	0
7.700	12.014.134.275.618.300	4.325.581.395.348.830	4.711.177.794.448.610	0.0	0.0	0.0	0	0.0	0	0	0	0
	0.0	0.0	0.0	16.0	5.0	11.0	0	0.0	0	0	0	0
48	2.579.505.300.353.350	17.209.302.325.581.300	3.945.986.496.624.150	19.0	10.0	9.0	0	1.0	0	0	0	0
	4.381.625.441.696.110	2.837.209.302.325.580	5.588.897.224.306.070	0.0	0.0	0.0	0	0.0	0	0	0	0
1.800	0.0	4.604.651.162.790.690	26.931.732.933.233.300	22.0	11.0	11.0	0	1.0	0	0	0	0
	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0	0	0	0
	0.0	0.0	0.0	22.0	11.0	11.0	0	1.0	0	0	0	0
8.110	9.787.985.865.724.380	10.0	10.0	22.0	11.0	11.0	0	1.0	0	0	0	0
	0.0	0.0	0.3075768942235559	21.0	10.0	11.0	0	1.0	0	0	0	0
	0.0	0.0	0.0	15.0	5.0	10.0	0	0.0	0	0	0	0
1.100	0.10600706713780919	8.372.093.023.255.810	2.393.098.274.568.640	21.0	10.0	11.0	0	1.0	0	0	0	0
	0.0	5.023.255.813.953.480	2.483.120.780.195.040	19.0	11.0	8.0	0	1.0	0	0	0	0
7.700	176.678.445.229.682	2.046.511.627.906.970	38.409.602.400.600.100	21.0	10.0	11.0	0	1.0	0	0	0	0
44	15.547.703.180.212.000	0.6046511627906976	132.033.008.252.063	6.0	2.0	4.0	0	0.0	0	0	0	0
5.470	21.201.413.427.561.800	0.13953488372093023	3.030.757.689.422.350	16.0	5.0	11.0	0	0.0	0	0	0	0
6	2.508.833.922.261.480	4.465.116.279.069.760	5.003.750.937.734.430	0.0	0.0	0.0	0	0.0	0	0	0	0
72	34.275.618.374.558.300	0.7906976744186046	3.675.918.979.744.930	18.0	10.0	8.0	0	1.0	0	0	0	0
9.430	3.957.597.173.144.870	0.0	4.388.597.149.287.320	10.0	5.0	5.0	0	0.0	0	0	0	0

Ilustración 29: Fichero de datos 2

	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL
2	Lab-Quincena-10-2	Teo-Quincena-11-1	Lab-Quincena-11-1	Teo-Quincena-11-2	Lab-Quincena-11-2	Teo-Quincena-12-1	Lab-Quincena-12-1	Teo-Quincena-12-2	Lab-Quincena-12-2	Asist	Ej1	Ej2	Ej3	Media ejercicios
	3.0	2.0	2.0	2.0	1.0	2.0	0.0	0	1.0	9.090.909.090.909.090		0.0	0.0	
	3.0	2.0	2.0	2.0	2.0	2.0	1.0	0	0.0	9.545.454.545.454.540	10.0	0.0	0.0	44.444.444.444.444.400
	3.0	2.0	2.0	1.0	2.0	2.0	1.0	0	1.0	9.090.909.090.909.090	7.5	7.5	10.0	9.444.444.444.444.440
	3.0	1.0	1.0	2.0	1.0	2.0	1.0	0	1.0	8.636.363.636.363.630		0.0	0.0	
	3.0	2.0	2.0	2.0	2.0	2.0	1.0	0	1.0	10.0	7.5	10.0	0.0	55.555.555.555.555.500
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0	7.5	7.5	10.0	9.444.444.444.444.440
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0	10.0	0.0	7.5	7.777.777.777.777.770
	3.0	2.0	1.0	1.0	0.0	2.0	0.0	0	0.0	6.818.181.818.181.810	7.5	0.0	10.0	7.777.777.777.777.770
	3.0	2.0	2.0	2.0	2.0	2.0	1.0	0	1.0	10.0	10.0	0.0	5.0	6.666.666.666.666.660
	2.0	2.0	0.0	2.0	2.0	2.0	0.0	0	0.0	72.727.272.727.272.700	10.0	10.0	10.0	11.111.111.111.111.100
	3.0	2.0	1.0	2.0	0.0	2.0	0.0	0	0.0	6.818.181.818.181.810	7.5	0.0	0.0	3.333.333.333.333.330
	3.0	2.0	1.0	1.0	2.0	1.0	1.0	0	1.0	8.636.363.636.363.630	10.0	0.0	10.0	8.888.888.888.888.890
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0	10.0	0.0	5.0	6.666.666.666.666.660
	3.0	2.0	2.0	2.0	2.0	1.0	1.0	0	1.0	9.545.454.545.454.540	10.0	0.0	0.0	44.444.444.444.444.400
	3.0	2.0	1.0	2.0	0.0	2.0	0.0	0	0.0	72.727.272.727.272.700	7.5	0.0	0.0	3.333.333.333.333.330
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0	10.0	10.0	10.0	11.111.111.111.111.100
	3.0	2.0	1.0	2.0	0.0	2.0	0.0	0	0.0	72.727.272.727.272.700		0.0	0.0	
	3.0	2.0	2.0	1.0	2.0	2.0	0.0	0	1.0	8.636.363.636.363.630	7.5	0.0	10.0	7.777.777.777.777.770
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0	10.0	10.0	10.0	11.111.111.111.111.100
	3.0	2.0	2.0	2.0	2.0	2.0	1.0	0	1.0	10.0	10.0	0.0	0.0	44.444.444.444.444.400
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0		10.0	5.0	
	3.0	2.0	2.0	2.0	2.0	2.0	1.0	0	1.0	10.0		0.0	0.0	
	3.0	2.0	2.0	2.0	2.0	2.0	1.0	0	1.0	10.0	7.5	10.0	10.0	10.0
	2.0	2.0	2.0	2.0	2.0	2.0	1.0	0	1.0	9.545.454.545.454.540		0.0	0.0	
	3.0	2.0	1.0	1.0	0.0	2.0	0.0	0	0.0	6.818.181.818.181.810		0.0	0.0	
	3.0	2.0	1.0	2.0	2.0	2.0	1.0	0	1.0	9.545.454.545.454.540	7.5	0.0	5.0	55.555.555.555.555.500
	3.0	2.0	2.0	0.0	2.0	2.0	1.0	0	1.0	8.636.363.636.363.630	7.5	5.0	10.0	8.888.888.888.888.890

Ilustración 30: Fichero de datos 3

6.2 Estructura de datos

Para finalizar este apartado, se muestra la estructura de directorios que conforman el proyecto. Como se puede apreciar, se mantiene el código de colores usado en el diagrama anterior.

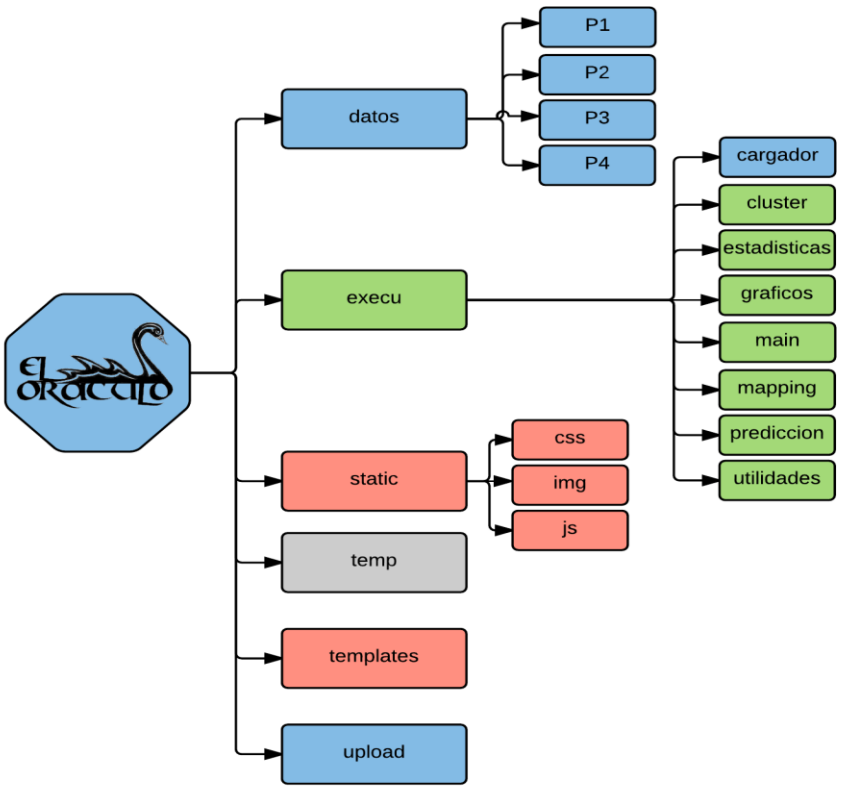


Ilustración 31: Estructura de ficheros

De esta forma, el directorio **"/datos"** almacenará los ficheros con información de los alumnos, así como los ficheros generados por el sistema con otros fines. Para que el profesor no tenga la obligación de introducir todos los tipos de ficheros, hacemos lo siguiente. Podemos ver que en este directorio existen cuatro subdirectorios, "P1", "P2", "P3", y "P4". En ellos se almacenarán los ficheros en función de aquellos que el profesor haya introducido:

- P1: Se almacenan aquí cuando el profesor introduce los ficheros de *accesos* y *calificaciones*.

- P2: Se almacenan aquí cuando el profesor introduce los ficheros de *accesos*, *calificaciones y ejercicios*.

- P3: Se almacenan aquí cuando el profesor introduce los ficheros de *accesos*, *calificaciones y asistencias*.

- P4: Se almacenan aquí cuando el profesor introduce los ficheros de *accesos*, *calificaciones, ejercicios y asistencias*.

Así mismo, el directorio **"/upload"** es utilizado como directorio auxiliar para la validación de los ficheros en el proceso de carga.

El directorio **"/execu"** almacenará la lógica de la aplicación distribuida en sus correspondientes subdirectorios. No obstante, aunque el subdirectorio **"/cargador"** pertenece a la lógica de la aplicación, representa la funcionalidad referente al tratamiento de datos. Es por esto que dicho directorio se encuentra en color azul.

Por último, el directorio **"/templates"** almacena todas las vistas, implementadas en html específicamente para el proyecto. Para la renderización correcta de los templates, obtendrá los recursos tales como hojas de estilos, código JavaScript e imágenes de los subdirectorios concretos en el interior del directorio **"/static"**.

6.3 Base de datos

En este apartado procederemos a describir el diseño e implementación de la base de datos. Además, hablaremos de la metodología empleada para el desarrollo de la misma. Para el desarrollo de la base de datos se han utilizado dos gestores de bases de datos distintos, MySQL y MariaDB. Aunque ambos sistemas son compatibles en la mayor parte de su funcionalidad, mediante el uso de ambos en paralelo, se asegura una compatibilidad total hacia el proyecto.

El objetivo de la base de datos es proporcionar una gestión de usuarios y permisos. Para esto, no sólo se guarda información sobre los usuarios y roles. Se guarda información sobre las asignaturas, grupos que se forman para dichas asignaturas, rutas de ficheros y la correspondiente relación entre las rutas de los ficheros y el grupo o asignatura al que

pertenezca.

Para mostrar todo esto, a continuación, se muestra un diagrama de la base de datos que se ha generado para tal efecto.

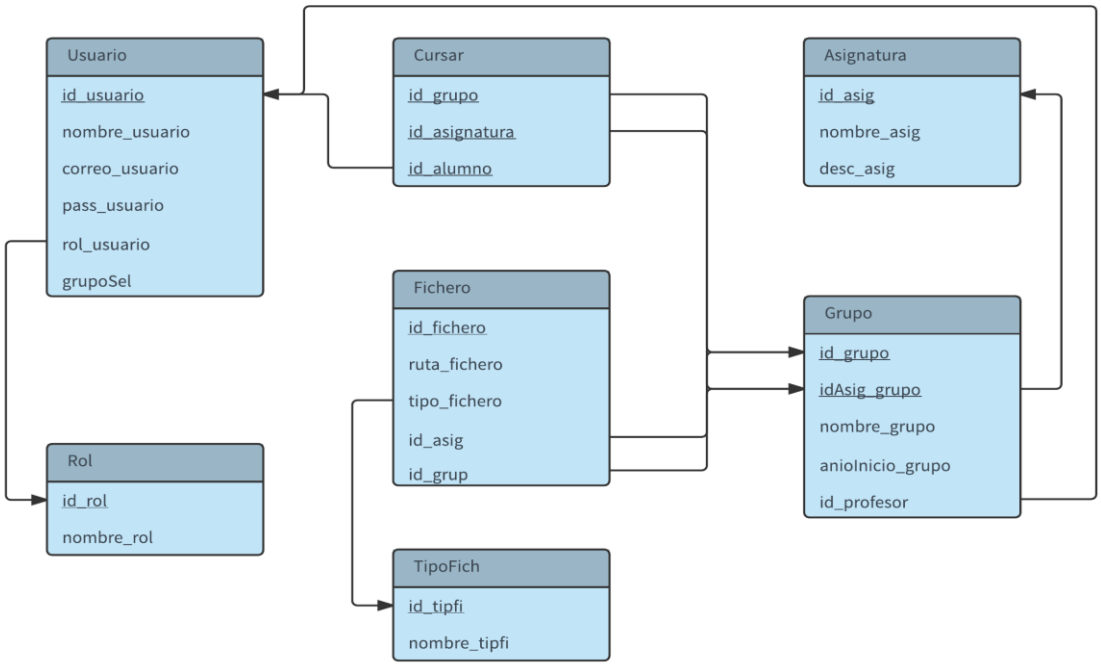


Ilustración 32: Diagrama ER de la base de datos

Nuestras entidades más importantes son el Usuario (con su Rol correspondiente), el Fichero, la Asignatura y el Grupo. El resto de tablas provienen de relaciones entre estas.

Como se puede observar en el diagrama, poseemos una tabla que almacena los datos básicos de un usuario. De estos, cabe destacar que el correo que se especifica es un valor con índice **"unique"** para asegurar un login único en el sistema. Un determinado usuario, posee un rol que, en el estado actual del sistema, puede ser el de **"Administrador", "Alumno"** o el de **"Profesor"**. Recordar que la funcionalidad del alumno no se encuentra desarrollada en esta versión de la herramienta.

Además de esto, guardamos información de una determinada asignatura, hasta ahora la asignatura descrita en múltiples ocasiones **(DSI)**. Para cada una de estas asignaturas, existen grupos concretos que imparten dicha asignatura. La información referente a cada grupo, es dependiente de la asignatura a la que se encuentra asociada. Es por esto, que se encuentra implementada como una entidad débil.

Cada usuario con el rol de profesor, puede impartir un determinado grupo mientras que dicho grupo tiene que tener un único profesor asociado obligatoriamente. Como se puede observar, existe una tabla intermedia denominada **"Cursar"** la cual tiene como objetivo vincular una serie de alumnos al grupo.

A continuación, mostramos una imagen de los tipos de datos de todas las columnas de la base de datos.

Table	Column	Type	Nullable	Extra
asignatura	id_asig	int(11)	NO	auto_increment
asignatura	desc_asig	varchar(2000)	YES	
asignatura	nombre_asig	varchar(300)	YES	
cursar	id_grupo	int(11)	NO	
cursar	id_asignatura	int(11)	NO	
cursar	id_alumno	int(11)	NO	
fichero	id_fichero	int(11)	NO	auto_increment
fichero	ruta_fichero	varchar(300)	YES	
fichero	tipo_fichero	int(11)	YES	
fichero	id_asig	int(11)	YES	
fichero	id_grup	int(11)	YES	
grupo	id_profesor	int(11)	YES	
grupo	anioInicio_grupo	int(11)	YES	
grupo	idAsig_grupo	int(11)	NO	
grupo	nombre_grupo	varchar(100)	YES	
grupo	id_grupo	int(11)	NO	auto_increment
rol	id_rol	int(11)	NO	auto_increment
rol	nombre_rol	varchar(100)	YES	
tipofich	id_tipfi	int(11)	NO	auto_increment
tipofich	nombre_tipfi	varchar(100)	YES	
usuario	pass_usuario	varchar(100)	YES	
usuario	rol_usuario	int(11)	YES	
usuario	id_usuario	int(11)	NO	auto_increment
usuario	nombre_usuario	varchar(100)	YES	
usuario	correo_usuario	varchar(100)	YES	
usuario	grupoSel	int(11)	YES	

Ilustración 33: Columnas de la base de datos

Capítulo 7: Visualización y Predicción

Capítulo 7: Visualización y Predicción

Para cumplir con los objetivos propuestos para el proyecto, es necesario analizar los datos de tres formas distintas: estadísticas, agrupamientos (o clustering) y predicción.

En la parte de estadísticas se realizará un análisis objetivo de los datos de cada una de las variables de estudio por separado.

En la parte de clustering, por otro lado, se agruparán los datos para sacar patrones de comportamiento relacionando las variables entre ellas.

Por último, se hará la predicción creando un modelo a partir del histórico de la información y contrastando los datos nuevos con ese modelo.

En este capítulo se profundizará en estos tres tipos de análisis.

7.1 Estadísticas

Con esta parte del módulo de análisis se pretende ofrecer al profesor una representación gráfica de la información que ha introducido al sistema. Con esto, puede ver en qué rango de notas se mueven sus alumnos o cómo está siendo la asistencia a sus clases.

Los cálculos estadísticos empleados en este submódulo son la media aritmética, la varianza y la desviación típica.

- Media: medida de centralización que se define como

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Varianza: medida de dispersión que se define como

$$v_x = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2$$

- Desviación típica: medida de dispersión que se define como

$$\delta = \sqrt{v_x}$$

Las fórmulas anteriores se pueden encontrar en *Estadística Aplicada* [19].

7.1.1 General

Como se puede ver en la siguiente imagen, primero se muestra una visión global de la situación académica del grupo.

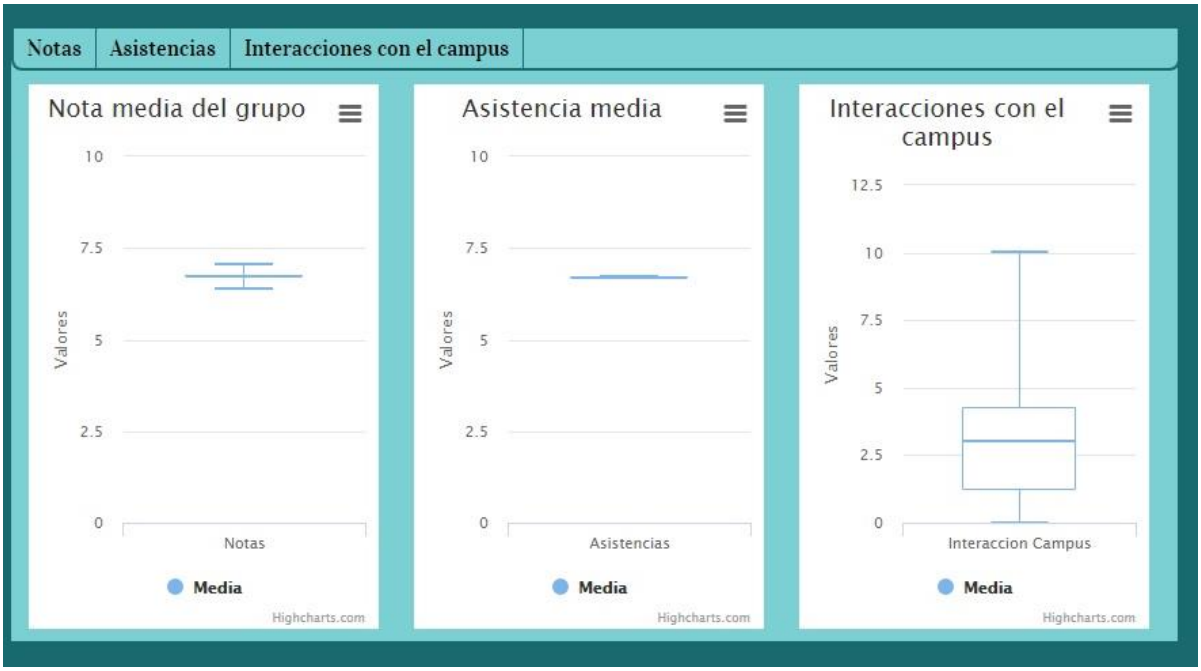


Ilustración 34: Gráficas generales

Para las notas se utilizan las calificaciones para hacer los cálculos y dibujar la gráfica. En caso de no disponer de los datos, la gráfica aparecerá vacía. En el grupo que se está analizando, la nota media ronda el 6 alto con variaciones entre el 6 y el 7.

Para las asistencias se calcula el ratio de asistencia sobre 10 de cada alumno para, posteriormente, hacer una media y una desviación típica. Como se puede observar. la desviación típica es muy escasa porque la asistencia es obligatoria mantenerla, como mínimo, al 80% para poder optar a una evaluación por proyecto. Aun así, la media está por debajo del 8. Esto puede deberse a alumnos que no asisten porque abandonan la asignatura o, por el contrario, a alumnos que deciden hacer el examen y, por tanto, van a menos del 80% de las clases.

Por último, están las interacciones con el campus virtual. Tomando como margen superior la cantidad de accesos del alumno que más interactúa con la plataforma, se realiza un ratio sobre 10 de la cantidad de interacciones por estudiante. Finalmente, se calcula la media y la desviación. En este caso, la media está muy baja, bastante por debajo del 5. Esto se deber a que la mayoría de los alumnos han realizado muchos menos accesos que el estudiante con el máximo.

7.1.2 Notas

En esta parte se encuentran representadas todas las notas de todos los ejercicios, prácticas, proyecto/examen y participación de la que se dispone. Para esta gráfica también se emplean los cálculos de la media y la desviación típica.

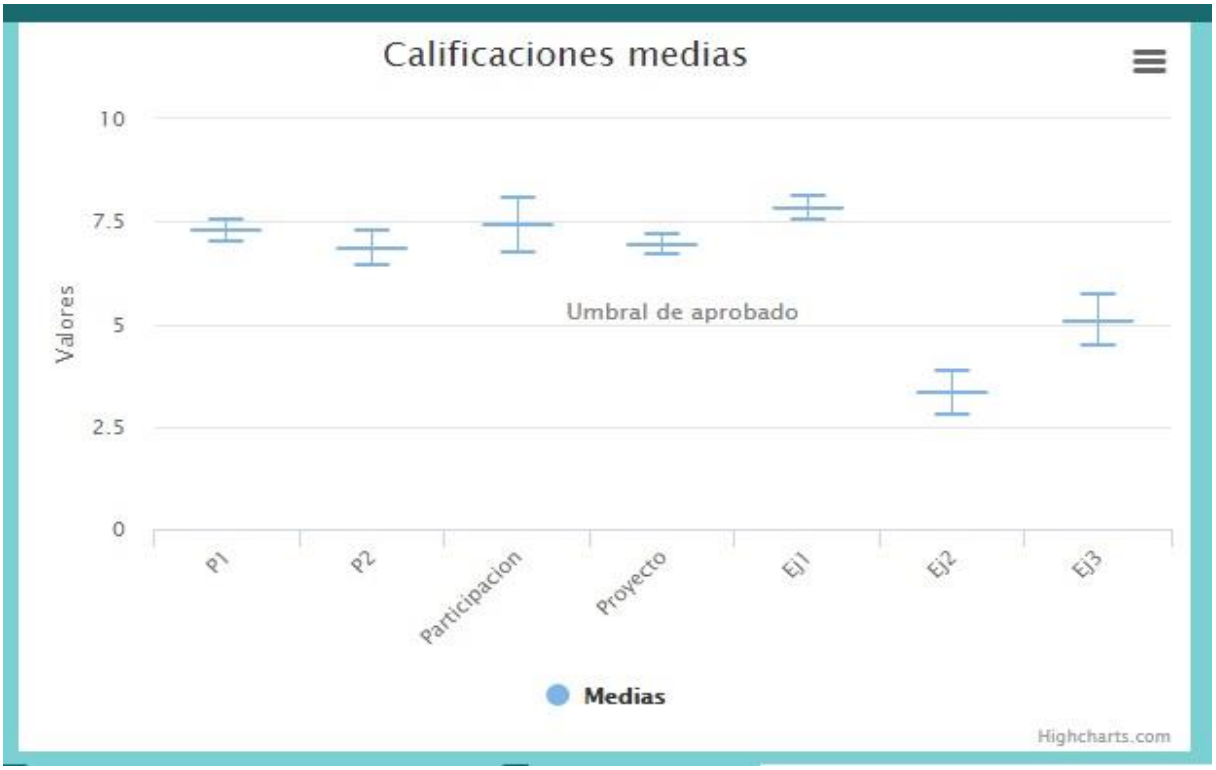


Ilustración 35: Gráfica de notas

Las prácticas son entregas parciales del proyecto final que se van realizando a lo largo del cuatrimestre. Por ese motivo, se puede observar una similitud bastante significativa entre las representaciones de P1, P2 y el proyecto.

Por otro lado, la participación es una media ponderada entre la asistencia, Ej1, Ej2 y Ej3. En este caso, sí se puede ver una variación más amplia debido, entre otras cosas, a la falta de obligatoriedad de los ejercicios.

7.1.3 Asistencias

En las gráficas que se presentan a continuación se muestran las asistencias totales. No se hace ningún tipo de media, sino que se cuenta cuántos alumnos van a clase cada día.

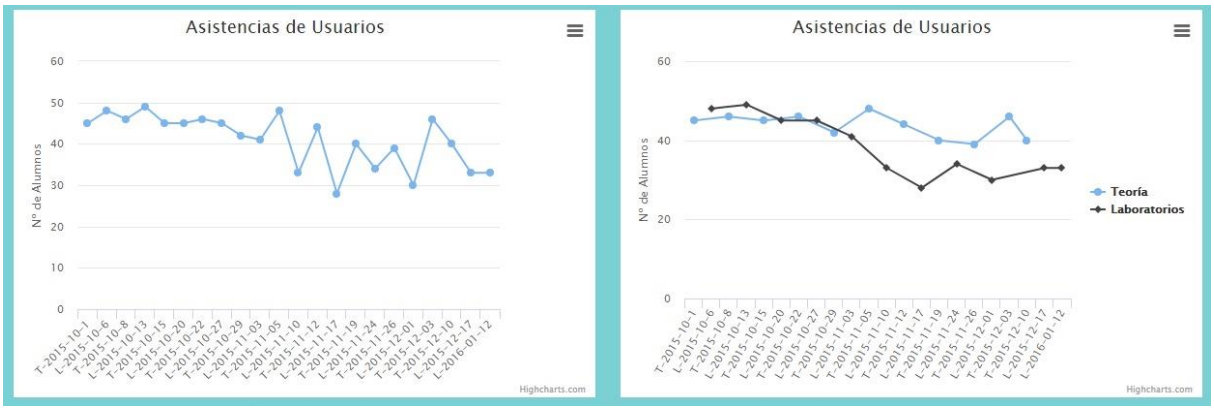


Ilustración 36: Gráficas de asistencias

En la gráfica de la izquierda se puede ver la asistencia de forma continua, sin discriminar si son clases de teoría o de laboratorio. En la de la derecha, por el contrario, sí se hace esa diferenciación. Esto permite al profesor comprobar cuál de los dos tipos de clase son más concurridas.

En la gráfica de la izquierda se puede ver que a partir de principios de noviembre la asistencia se vuelve irregular. Sin embargo, comparada con la de la derecha, se observa que la asistencia a clase de teoría se mantiene estable y que son las clases de laboratorio las que van descendiendo en asistentes.

7.1.4 Interacciones con el Campus Virtual

En esta gráfica aparecen las medias de accesos por día junto con sus desviaciones típicas. La línea de tonalidad azul claro representa la media y el sombreado que la envuelve representa su desviación típica. Cada una de estas medias se hace sumando todos los accesos de un determinado día y dividiendo entre el número de estudiantes que interaccionan con el campus dicho día.

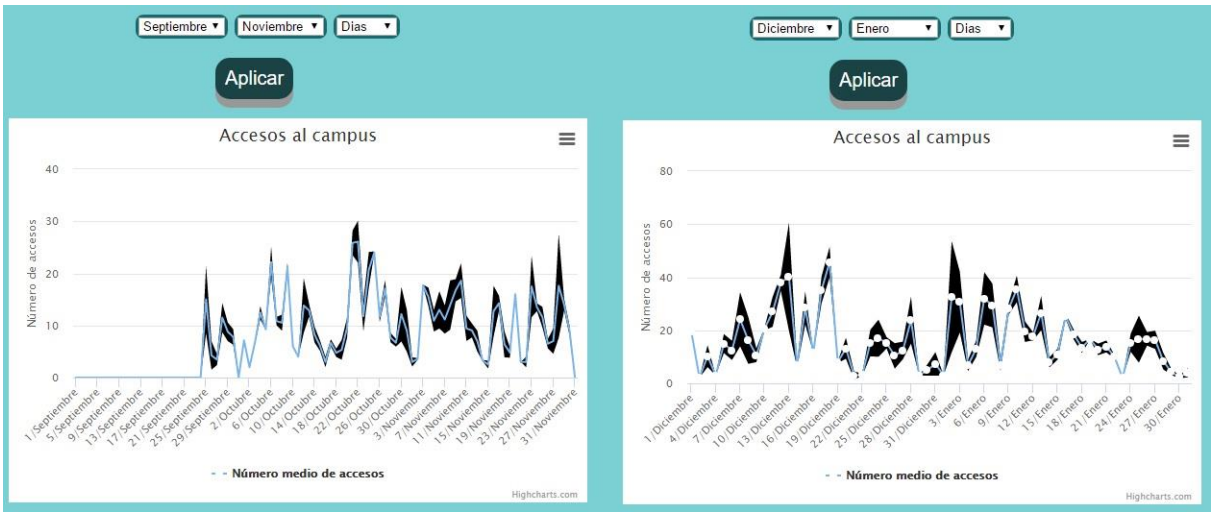


Ilustración 37: Gráficas de accesos al campus virtual

La ilustración 37 corresponde con todos los accesos hechos por un grupo durante el cuatrimestre, es decir, de septiembre a finales de enero. En la imagen de la izquierda se representa la primera mitad de dicho intervalo (septiembre-noviembre); y en la de la derecha, la otra mitad (diciembre-enero). Como se puede observar en las gráficas, en general la media de accesos diarios en la segunda mitad del curso, que oscila entre 0 y 50, duplica la de la primera, que varía entre 0 y 26 aproximadamente. Además, hay picos de interacciones, que pueden revelar fechas de entrega de ejercicios, prácticas o el proyecto final.

7.2 Clustering

El objetivo de este módulo del sistema es obtener una serie de agrupaciones con los datos de cursos anteriores cargados previamente. Cada una de estas agrupaciones corresponde con un modelo de comportamiento o rendimiento académicos. Hacer esto nos permite encuadrar a cada uno de los alumnos actuales dentro de uno de los grupos. De esta forma podemos observar la trayectoria que posee y clasificarlo en un modelo de conducta de los obtenidos previamente. Así obtenemos una visión general sobre la actitud hacia la asignatura que dicho alumno posee.

En función del tipo de resultados que el usuario, en este caso el profesor, desee obtener, puede recurrir a un método de agrupamiento jerárquico o no jerárquico [20].

En el primero de ellos, los datos se agrupan consecutivamente siguiendo un esquema jerárquico. Para ello, se utiliza una medida entre dos pares de datos. Dicha distancia lógica suele venir dada por el cálculo de distancias euclídeas teniendo, como dimensiones a tener en cuenta, las variables que se hayan decidido utilizar para el proceso de clustering. No obstante, existen otro tipo de distancias utilizadas para tal efecto.

El segundo, agrupa los datos por cercanía en un espacio de coordenadas con tantos ejes como atributos tengan los datos. Este método sólo sirve para trazar patrones de comportamiento.

Sobre ambos se profundizará más adelante en este mismo capítulo.

7.2.1 Pasos previos al clustering

Aunque el proceso interno de clustering que podemos ver en ambos es diferente, la interacción con respecto al usuario y los pasos previos son similares. Estos pasos previos corresponden a la adaptación de los datos de entrada.

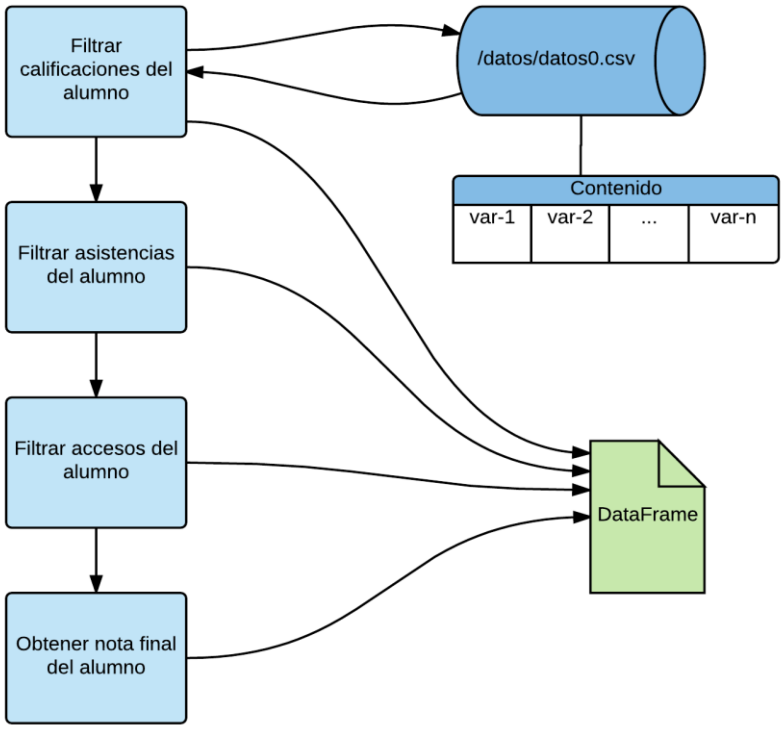


Ilustración 38: Proceso de generación del fichero de datos

En ambos casos se parte de un archivo en el que se encuentran almacenados los datos completos de cursos anteriores. De dicho fichero se emplean una serie de variables pertenecientes a cada alumno para crear los diferentes clusters. Todas las fases de este proceso van almacenando la información obtenida en un DataFrame de la librería de Pandas, que será el que usen los algoritmos de clustering.

Primero se filtra toda la información correspondiente a las calificaciones generadas para cada alumno (a excepción de la nota del examen o el proyecto) y se tratan convenientemente, modificando su escala de 0 a 10. De esta forma quedan en una escala uniforme con el mismo peso en los algoritmos sucesivos. Posteriormente, se hace una media con todas ellas y se obtiene la variable que denominamos **Evaluación continua**.

Siguiendo el mismo concepto empleado para la obtención de la variable **Evaluación continua**, se procede a la obtención de la variable **Asistencia.** Esto se realiza mediante el cálculo del ratio de presencia en clase. Para ello, transformamos la escala a una escala sobre 10 con el objetivo de que se ajuste adecuadamente a la calculada previamente.

Para continuar, con las interacciones al campus virtual se procede de forma similar. Tomamos como margen superior el número de accesos del alumno que más veces ha entrado en la plataforma. Dicho estudiante tendría un 10 en **Accesos** y los demás irían en relación a este.

La última variable de interés, que es importante mencionar en este apartado, es la **Nota del Examen o Proyecto**, a la que no habría que tratar de ninguna forma.

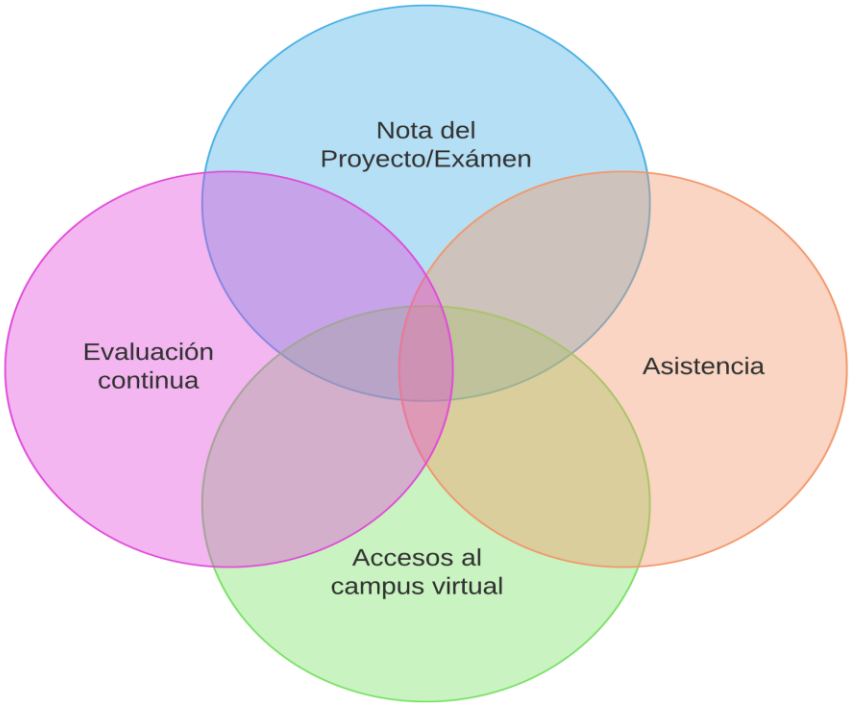


Ilustración 39: Variables utilizadas para el clustering

Hecho esto, las variables formarían una tupla de 4 valores por cada alumno que haya cursado la asignatura. De esta forma, se crea una colección de datos a raíz de las tuplas que se acaban de mencionar. Estas pasan a los algoritmos de clustering para obtener los clusters en los que se encuentran los alumnos.

7.2.2 Clustering No Jerárquico

Como ya se mencionó previamente, este tipo de clustering agrupa los datos por proximidad. Para crear estos clusters existen varios algoritmos diferentes. Tras analizarlos, se ha decidido usar el algoritmo KMeans debido a que, el procesamiento interno del mismo, hace las divisiones en torno a centroides. Esto facilita la posterior interpretación de los resultados obtenidos.

En el proceso de decisión de las herramientas a utilizar y de análisis de los datos, se consideró el uso de numpy para el procesamiento del algoritmo KMeans. No obstante, quedó descartado debido a que la implementación de dicho algoritmo en esta librería devuelve distancias euclídeas en lugar de los centroides, haciendo excesivamente complicada la interpretación de los resultados. Es por esto que, para el desarrollo de esta parte, se ha decidido el uso de la implementación que se encuentra en la librería *sklearn.cluster* [21], gracias a que éste devuelve los centroides de cada grupo, los cuales son más cercanos al tipo de calificaciones que se suelen incorporar en el aula y, por tanto, es

más fácil para el profesor analizar los resultados obtenidos.

El resultado del algoritmo KMeans [22] suele representarse como una partición del espacio de datos en celdas de Voronoi, también conocidos como polígonos de Thiessen. Cada una de las celdas corresponde con un cluster concreto, en cuyo interior se encuentran los datos pertenecientes al grupo. A continuación, y a modo de ejemplo, se presenta una imagen con la representación de un conjunto de clusters.

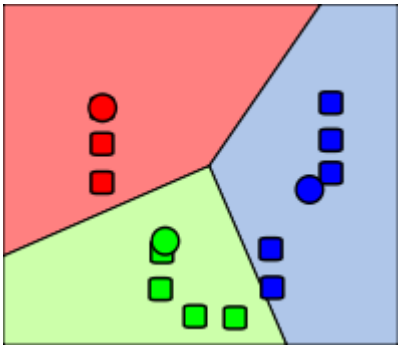


Ilustración 40: Ejemplo de clustering no jerárquico [23]

En la imagen se pueden ver tres polígonos con un conjunto de cuadrados y un círculo en el interior de cada uno de ellos. Los polígonos son cada uno un cluster, los cuadrados los datos que lo forman y el círculo, el centroide.

Esto se define mediante refinamiento iterativo, cuyos pasos, grosso modo, son:

1. Definir unos centroides aleatorios iniciales
2. Agrupar cada dato en torno al centroide más cercano. Cada uno de los datos es una tupla de n dimensiones.
3. Recalcular unos nuevos centroides a partir de los datos agrupados.
4. Repetir los pasos 2 y 3 hasta que no se produzcan más modificaciones en los centroides.

Tras realizar los pasos anteriores, se obtienen los clusters con sus correspondientes centroides. Esto corresponde con la implementación clásica del algoritmo, también conocido como **algoritmo de Lloyd**.

Este algoritmo es computacionalmente difícil, pero la gran cantidad de heurísticas empleables hace que su complejidad varíe de unas a otras. La implementación elegida para el proyecto tiene una complejidad media de **O**(knT), donde k es el número de clusters, n es el número de elementos a clasificar y T es el número de iteraciones. En el caso peor, la complejidad podría llegar a **O**(n^(k+2/p)), siendo p el número de valores que forman la tupla dato [24].

Saber a qué grupo pertenece a cada dato se calcula con distancia euclídea, pues es la usada por defecto en la mayoría de los sistemas.

Una de las características del algoritmo es que permite elegir el número de clusters que se

quieren crear, que en la implementación utilizada es mediante el parámetro n_clusters. En el presente proyecto, este parámetro se ha inicializado a 4 para que se aproxime, en la medida de lo posible, a las cuatro calificaciones esperadas: aprobado (AP), notable (NT), sobresaliente (SB) y suspenso (SS).

En caso de que el profesor no esté de acuerdo en el número de clusters predefinidos, se le proporciona un formulario para indicar el número que desee entre 1 y el número de alumnos utilizados para el procesamiento. De esta forma, el profesor podrá obtener una visión de los modelos de conducta desde un punto de vista más o menos preciso en función del número de conductas que desee obtener.

En el desarrollo de la aplicación, se ha decidido representar este tipo de clustering de dos formas diferentes: con una gráfica poligonal o una gráfica lineal. En ellas, cada cluster está asociado a un color diferente. Esto ocurre por defecto, aunque los desarrolladores pueden, en cualquier momento, especificar dicho rango de colores como deseen.

A continuación, se muestra una gráfica que extraída del presente proyecto con objeto de proporcionar al lector un ejemplo de lo anteriormente mencionado.

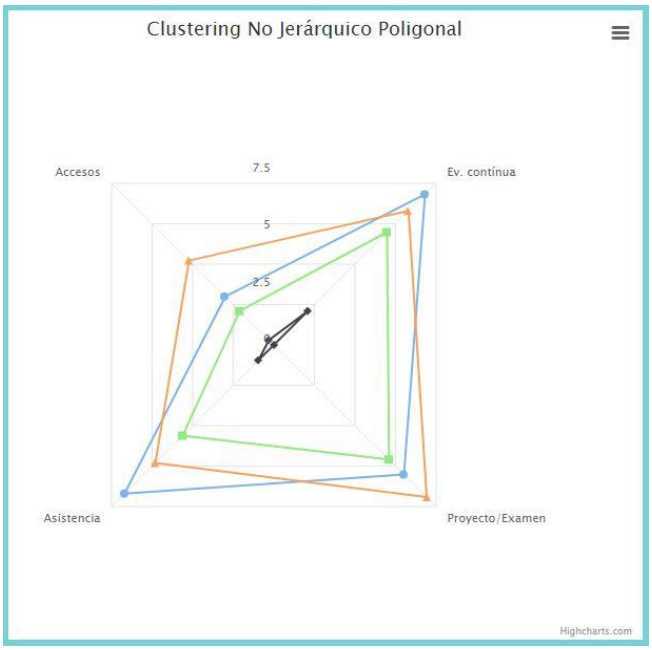


Ilustración 41: Gráfica de grupos de alumnos con conductas similares (poligonal)

En esta primera gráfica, se puede observar cómo cada cluster va envolviendo al anterior, a excepción de los dos últimos, que se entrecruzan. Esta representación permite al profesor analizar los comportamientos más habituales y la relación que hay entre ellos, sin centrarse tanto en los valores numéricos.

A modo de ejemplo, vamos a centrarnos en el cluster negro, en el cual, teniendo en cuenta el bajo número de accesos y la nota proyecto/examen tan próxima cero, se puede llegar a la conclusión de que este cluster está formado por alumnos que han abandonado la asignatura.

Si continuamos expandiéndonos hacia el exterior, el cluster verde parece formado por alumnos que muestran una actitud ligera hacia a la asignatura, buscando aprobar la misma sin excesivo esfuerzo en ella.

Por último, si nos centramos en los clusters azul y naranja, se pueden inferir dos relaciones: una entre el número de accesos y la nota del examen o proyecto, y otra entre la evaluación continua y la asistencia. Esto puede ser debido a que los alumnos que más acceden al campus virtual realizan un proyecto más correcto en cuestión de teoría y formato, llevando esto a conseguir una nota mayor en el mismo. Por el contrario, una buena asistencia a las clases hace que el estudiante sea más consciente de las fechas de entrega y consejos acerca de los ejercicios que se van entregando durante el curso.

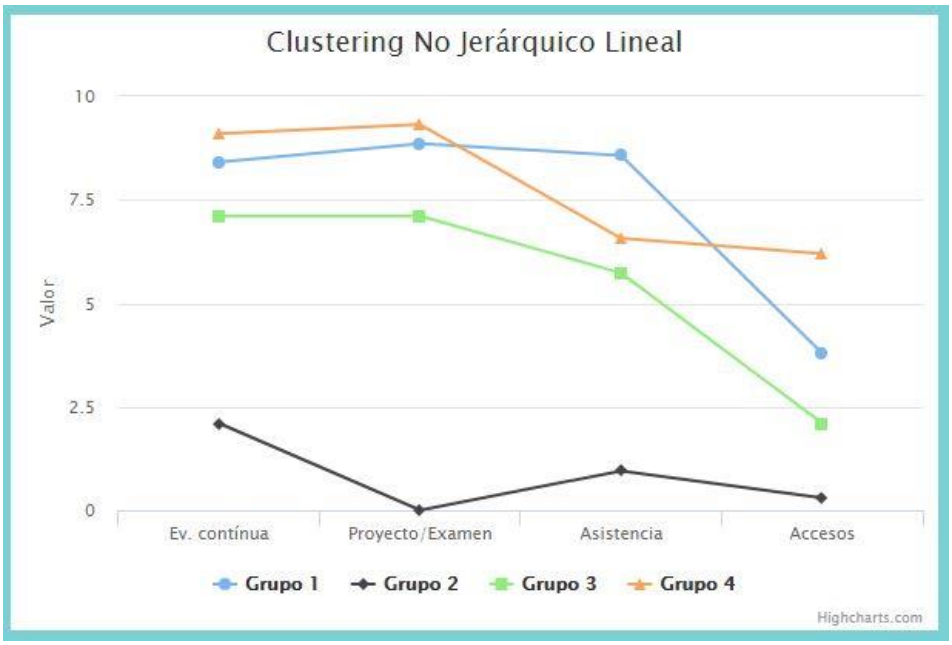


Ilustración 42: Gráfica de grupos de alumnos con conductas similares (lineal)

A continuación, podemos ver una segunda visualización del resultado de aplicar KMeans. Esta está hecha en un momento diferente al de la gráfica anterior, pero con los mismos datos. Los resultados obtenidos son ligeramente diferentes porque los centroides iniciales de esta segunda ejecución de KMeans son distintos a los de la primera. Esto se debe a que el algoritmo realiza una pre-ejecución del algoritmo EM (Esperanza-Maximización), que define una serie de clusters a través de una Función de Densidad de Probabilidad. Esto lo realiza en dos etapas, esperanza y maximización, que repite hasta alcanzar un resultado. Los centroides de los clusters calculados por EM [25] se convierten en los centroides iniciales que utiliza KMeans.

La implementación de KMeans utilizada en el proyecto cuenta con un parámetro **algorithm**, el cual define qué tipo de EM se lleva a cabo. Esto es definible por el programador, aunque debe tenerse en cuenta la densidad de los datos, pues hay modos que no soportan la ejecución sobre datos densos. En este caso se está utilizado en modo “automático”, lo que significa que si son densos se usa la variación “elkan” o, si están dispersos, se utiliza la

implementación clásica.

En esta segunda ejecución del clustering no jerárquico se puede observar que las conclusiones sobre los grupos negro y verde se mantienen respecto al gráfico poligonal. Sin embargo, hay ligeras variaciones en los grupos naranja y azul.

Por cómo se entrecruzan las líneas, se vuelve a presentar el patrón de que aquellos alumnos que acceden más al campus virtual, también son los que mejores notas sacan en el proyecto o examen. Por otro lado, los que más asisten a clase no son, necesariamente, los que mejor calificación sacan en la evaluación continua.

También se puede observar cómo los estudiantes de grupo azul son bastante constantes en su comportamiento y calificaciones a lo largo del curso. La única variable que se sale del patrón son los accesos. Sin embargo, esto no es realmente significativo ya que no se trata de una calificación, sino de un ratio en función del alumno que más accesos realiza. Esto puede descompensar el valor de la variable si dicho alumno es un caso límite que tiene un excesivo número de accesos en comparación con el resto de estudiantes

En caso de que el profesor quiera información más concreta, debajo de la gráfica hay un desplegable que contiene en qué cluster se sitúa cada alumno.

7.2.3 Clustering Jerárquico

En este método, utilizamos una metodología de agrupamiento jerárquico aglomerativa. Se asume que, al inicio del algoritmo, cada elemento es un grupo en sí mismo y que, con cada iteración, se van agrupando por pares según un cierto grado de afinidad. El algoritmo finaliza cuando se consigue un solo grupo compuesto por el resto de ellos. De esta forma, queda una estructura jerárquica a modo de árbol.

El clustering jerárquico tiende a representarse comúnmente en un dendrograma que muestra a la perfección todo el proceso iterativo. A continuación, se muestra una gráfica de un dendrograma a modo de ejemplo representativo.

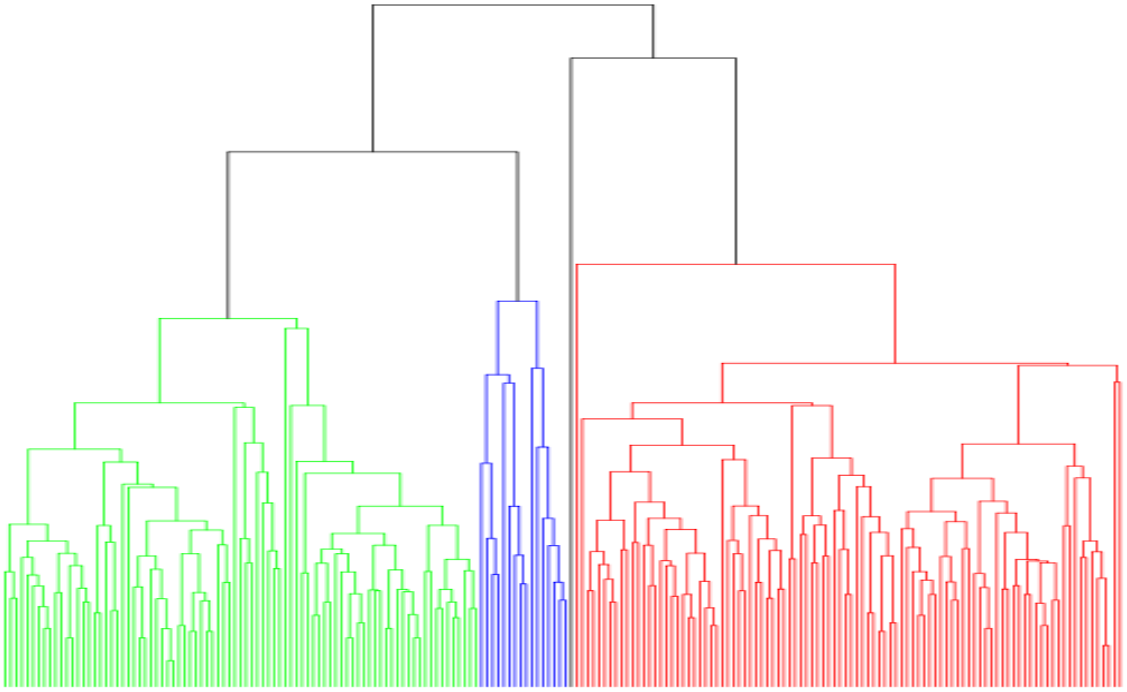


Ilustración 43: Dedrograma [26]

Como se puede observar, en el eje de las **Xs** se encuentran los individuos de los que se pretende realizar el proceso de clustering. Estos se van agrupando de forma sucesiva hasta que, finalmente, se obtiene un único grupo. El eje de las **Ys** nos indica la distancia lógica entre los individuos o grupos que se unen en esa iteración.

El uso de una metodología aglomerativa es muy importante de cara a la eficiencia del sistema puesto que la complejidad de este algoritmo es de $\mathbf{O(n^3)}$ siendo **n** el número de alumnos que se encuentren en el histórico de la asignatura. Esto puede parecer una complejidad demasiado elevada. No obstante, su metodología jerárquica opuesta, la metodología divisiva, posee una complejidad de $\mathbf{O(2^n)}$, lo cual nos llevaría a una situación considerablemente peor con grandes cantidades de datos.

La decisión de qué par de grupos se unen en una iteración concreta, viene definido por aquellos cuyo grado de afinidad sea mayor. El grado de afinidad es la distancia lógica entre pares de alumnos. En función del problema concreto que se desee abordar, se pueden utilizar distintos conceptos de distancia. A continuación, proporcionamos una lista de algunas métricas utilizadas a modo de distancia lógica:

- ***Distancia euclídea*** [27]***:*** Esta distancia es el concepto de distancia que suele usarse por defecto en cualquier sistema.

$$\boldsymbol{d_E(P_1, P_2) = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}}$$

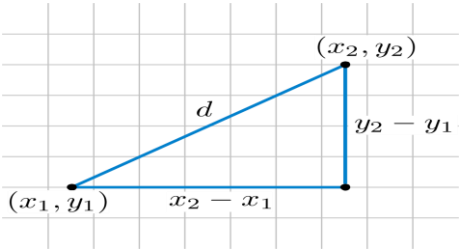


Ilustración 44: Distancia euclídea [46]

- ***Distancia máxima:*** Esta métrica consta de agrupar los distintos teniendo en cuenta

aquellos que se encuentren más alejados entre sí.

- ***Similitud de cosenos:*** Esta métrica consiste en la comprobación del coseno del ángulo formado entre dos vectores, estando comprendido en un rango [-1,1] siendo -1 cuando ambos vectores apuntan a distancias totalmente opuestas, 0 en el caso de que sean ortogonales y así sucesivamente.

- ***Distancia Manhattan*** [28]***:*** Como sustituto de la distancia euclídea, este tipo de distancia está basado en el concepto de distancia entre dos puntos en la mayor parte de las calles de Manhattan. El siguiente gráfico muestra que los extremos de la línea verde (línea que corresponde con la distancia euclídea) son los puntos representados en nuestro problema y el resto de líneas son las distancias en concepto de distancia Manhattan. El punto más importante a destacar y que resulta de interés en muchas situaciones, es que todas estas distancias (a excepción de la euclídea) son iguales.

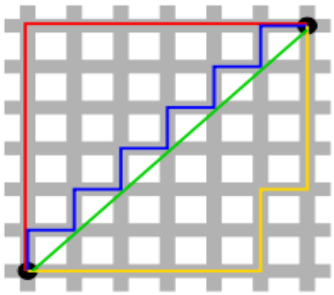


Ilustración 45: Distancia Manhattan [47]

Por desgracia, la mayor parte de estos tipos de distancias presentan el inconveniente de proporcionar resultados de disimilaridad redundantes cuando las variables están correlacionadas. Esto causa que se altere el correcto proceso de agrupamiento. Debido a que, en el presente proyecto, las variables presentan una fuerte correlación entre ellas, se ha decidido sustituir el uso del concepto de distancia descrito previamente, por el uso de tipos de distancia de similaridad.

Las distancias de similaridad utilizan coeficientes para mostrar el grado de similitud entre individuos. De esta forma reducimos considerablemente la redundancia mencionada previamente. Para esto, la librería scipy [29], nos proporciona un parámetro con la opción 'correlation' que sustituye el uso de la distancia común por un coeficiente con la siguiente fórmula:

$$1 - \ (u - \bar{u}) * (v - \bar{v}) / \left|\left|(u - \bar{u})\right|\right|_2 * \left|\left|(v - \bar{v})\right|\right|_2$$

Hasta aquí, hemos definido la métrica de comparación utilizada para este proyecto, así como el motivo por el cual dicha métrica se ha puesto en práctica en el mismo. Ahora nos queda definir el método de clustering utilizado para el agrupamiento de los individuos en función de la métrica utilizada.

Al margen de la métrica que se desee emplear en la comparación entre grupos, existen una serie de criterios con los que se pueden formar los grupos. A continuación, exponemos algunos que son frecuentemente utilizados:

- ***Distancia mínima:*** Se agrupan los individuos (o grupos ya formados en una iteración previa) que posean el valor mínimo para la métrica utilizada.

- ***Distancia máxima:*** Se agrupan los individuos (o grupos ya formados en una iteración previa) que posean el valor máximo para la métrica utilizada.

- ***Distancia entre centroides:*** Se agrupan los individuos (o grupos ya formados en una iteración previa) teniendo en cuenta el valor obtenido en la métrica empleada utilizando los centroides como puntos a comparar.

En este proyecto, se ha decidido el uso de la distancia mínima entre los individuos o grupos con el uso de una métrica de correlación.

Cabe destacar que, en este proyecto, se ha decidido realizar tres procesos de clustering, solapando los resultados de los mismos para obtener la gráfica que trataremos a continuación.

Es por esto que, aunque normalmente este tipo de resultados tienden a representarse a través de un dendrograma, en nuestra aplicación lo representamos a través de una gráfica circular. Dicha gráfica tendrá distintas capas con el objetivo de proporcionar una simplificación visual de agrupamiento.

En el siguiente gráfico podemos ver un ejemplo, sacado de la propia aplicación, de la gráfica.

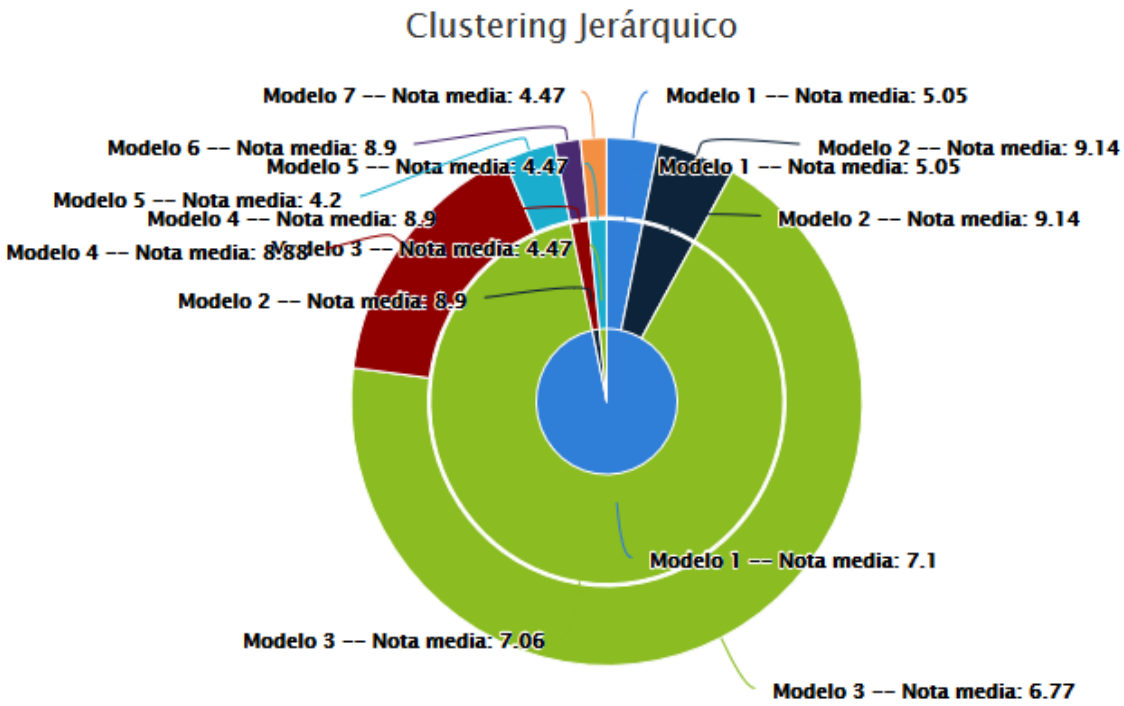


Ilustración 46: Gráfica de clustering jerárquico circular

Como se puede observar, en el ejemplo nos encontramos con una gráfica circular en tres. Cada capa posee el conjunto completo de los datos de los alumnos que se han sometido al proceso de clustering, es decir, cada capa representa un proceso de clustering distinto.

Si nos fijamos, la capa interna de la gráfica, podemos ver que se divide en tres grupos distintos, comenzando por un grupo grande de color azul y seguido por otros dos grupos pequeños.

El grupo más grande, representado por el color azul, está etiquetado como **"Modelo 1 -- Nota media: 7.1"**. Esto nos indica que este grupo es el primer modelo de conducta, obtenido en esa capa o nivel (la capa interna), y que, como se puede apreciar, está especificada para que obtenga un número de tres clusters. Además, nos indica que los integrantes del grupo mencionado anteriormente, poseen una nota media de 7.1.

Si pasamos a ver el segundo nivel (la capa intermedia de la gráfica), podemos ver que este proceso de clustering está específicamente configurado para que se obtengan cinco grupos. No obstante, lo más importante a tener en cuenta, es que está diseñado para que los alumnos incluidos en cada capa se encuentren distribuidos en el mismo espacio. Un ejemplo de esto lo podemos ver si volvemos a los dos agrupamientos del primer nivel (capa interna). Los alumnos incluidos en el interior de dichos agrupamientos, son los mismos que se encuentran incluidos en sus homólogos en capas superiores, propagándose a la parte superior de la gráfica.

De esta forma, haciendo una comparativa entre la capa interna y la capa intermedia, podemos decir que los integrantes del primer modelo de conducta obtenido en la capa interna, con una nota media de 7.1, se dividen en tres modelos de conducta en la capa intermedia con notas medias de 5.05 y 9.14 respectivamente.

Con este proceso obtenemos una visión más global sobre ambas capas comprobando que, cuanto mayor sea el número de cluster que se deseen obtener mayor nivel de precisión tiene. Es por esto que, volviendo al ejemplo, nos damos cuenta que al aumentar el grado de refinamiento del algoritmo subiendo un nivel en la gráfica, hemos separados los casos extremos del cluster en dos clusters diferenciados.

Gracias a esta funcionalidad, el profesor es capaz de obtener información visual inmediata de la distribución de los modelos de conducta con respecto a los resultados finales que suelen producir dichos modelos de conducta. Además, debajo de la gráfica dispone de un desplegable distinto para cada uno de los niveles y que contiene información sobre a qué grupo pertenece cada alumno.

7.3 Sistema predictor

El objetivo de módulo del sistema consiste en proporcionar al profesor una predicción concreta de las calificaciones que sacarán sus alumnos en función de la evolución de los mismos hasta la fecha en la que el profesor alimentó a la herramienta con los datos necesarios para tal efecto.

Con objeto de introducir el sistema de predicción y analizar el comportamiento en este ámbito, en esta primera fase del proyecto se han empleado los métodos de árboles de decisión y de regresión lineal para realizar la predicción como se verá más adelante.

Para saber si estos algoritmos son adecuados y cumplen con lo que se esperar, se han realizado cuatro pruebas con cada uno de ellos, utilizando la siguiente información:

- La primera, con las notas del ejercicio 1 y de la práctica 1, las asistencias y los accesos al campus virtual.
- La segunda, con todo lo anterior y la nota del ejercicio 2.
- La tercera, con todo lo anterior y las notas de la práctica 2 y el ejercicio 3.
- La cuarta, con todo lo anterior y la nota del proyecto/examen.

Puesto que no se disponía de un histórico de otros años, se han utilizado los datos un solo curso anterior para entrenar y validar el sistema. Se han utilizado 48 sujetos para el entrenamiento y se han utilizado 61 sujetos para la creación de los modelos.

7.3.1 Funcionalidad

Partiendo de la colección de datos completos que se poseen de cursos pasados, se procede a construir modelos usando técnicas de aprendizaje supervisado sobre los que se pueda predecir la nota final con la mayor tasa de aciertos posibles [30].

Tanto en árboles de decisión como en regresión lineal, los datos utilizados son los mismos: todos aquellos ficheros que introduzca el profesor en la herramienta hasta un determinado momento. Estos datos son relativos a la asistencia, las interacciones con el campus virtual y las calificaciones de prácticas y ejercicios.

Para comenzar con el flujo de ejecución que acabamos de mencionar, primero se lee el fichero con toda la información de años anteriores y se cargan todos los datos en un DataFrame (estructura de datos proporcionada por la librería pandas). Acto seguido, con la información aportada por el profesor sobre el grupo que se quiere predecir, se crea otro DataFrame. Se comparan las columnas de las dos colecciones de datos y se eliminan del primer DataFrame especificado todas aquellas que no se encuentren en el segundo. En un tipo de datos Serie se almacenan las calificaciones finales obtenidas por los alumnos del histórico de la asignatura. Esta serie y el primer DataFrame especificado, se utilizan como variables, que se le pasan al algoritmo de predicción elegido, para obtener un modelo.

Una vez el modelo ha sido correctamente formado, se realiza una operación de predicción con cada uno de los estudiantes no calificados, obteniendo una nota esperada en un rango predefinido de cuatro valores [Suspenso, Aprobado, Notable, Sobresaliente].

En caso de disponer de las calificaciones finales reales de los alumnos a predecir, se puede obtener una tasa de aciertos del modelo generando un gráfico, correspondiente con una

matriz de confusión, que nos permite ver el grado de desviación que tiene nuestro algoritmo.

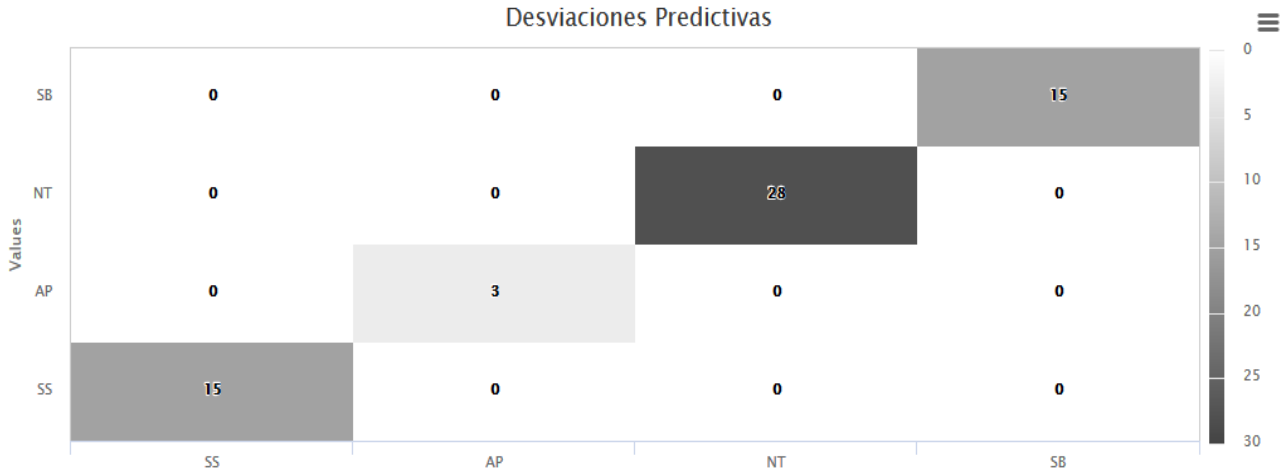


Ilustración 47: Matriz de confusión

A modo de ejemplo de lo anteriormente mencionado, observamos que, en la imagen anterior, sacada del presente proyecto, el eje de las **Xs** representa el valor que nuestra herramienta predice que va a tener la calificación de un alumno mientras que el eje de las **Ys** representa el valor real que los alumnos han sacado en su nota final. De esta forma, podemos ver que el algoritmo predice que 15 alumnos van a suspender y que, efectivamente, suspenden. Si seguimos con el ejemplo, se puede observar que todos los valores distintos de **0** se encuentran en la perpendicular de la gráfica.

Esto nos hace ver que la tasa de aciertos es un cien por cien ya que todos los valores de las **Xs** corresponden con los valores de las **Ys**. Como ya se ha comentado, esta gráfica es especialmente útil para analizar hacia qué lado se desvía el algoritmo de predicción utilizado, de esta forma, poder ajustarlo cuando sea necesario.

El ejemplo que se acaba de presentar es un caso de ideal de predicción, no obstante, en el estado actual del proyecto es usual encontrar pequeñas variaciones de uno o dos elementos hacia la parte superior de la gráfica. Este alto porcentaje de aciertos se debe, en gran medida, a las características tan peculiares de la asignatura en la que se ha centrado este proyecto.

Volviendo, nuevamente, a las posibilidades que se le muestra al profesor, también se le permite acotar la predicción en el tiempo a través de un formulario a modo de dos desplegables que se encuentran en la interfaz. Esto permitiría al profesor, por ejemplo, trabajar sólo con los datos del primer mes de clase aún a pesar de que se disponga de los datos completos de un cuatrimestre.

Toda la información obtenida (predicciones y tasa de aciertos, en caso de tenerla) se envía al servidor, que se encarga de darle formato y enviársela al entorno web.

7.3.2 Árboles de decisión

Se eligió el uso de este tipo de predicción porque se ajusta a las necesidades de sistema. Los árboles de decisión trabajan con un conjunto de estados objetivos discreto, que en caso de este proyecto se traduce en cuatro posibles calificaciones objetivo (aprobado, suspenso, notable y sobresaliente) a los que se llega a través del análisis de las distintas variables, creando un conjunto de nodos que contienen la información relativa a cada estado.

Se utiliza la implementación de *sklearn.tree*. Este algoritmo cuenta con la opción de elegir el número de características que se combinan y almacenan en cada nodo. Por defecto se guardan todas, pero el usuario puede decidir si usar una cantidad relativa a la raíz cuadrada o al logaritmo en base 2 del número total de variables.

Para la implementación de los árboles de decisión hay cuatro algoritmos: ID3, C4.5, C5.0 y CART. La librería sklearn desarrolla su código a partir de CART, que es una técnica que produce árboles de decisión o de regresión dependiendo de si los estados finales son cualitativos o continuos.

Cada nodo cuenta con una serie de reglas que lo hacen cierto o no. Estas reglas se generan a partir de reglas de asociación [31] que, con una base conocimiento grande, sacan conclusiones sobre hechos comunes. Esto puede llevar a que se creen una gran cantidad de reglas, pero no todas son fiables o ciertas. Para filtrar las reglas útiles, existen dos umbrales: **soporte** (support) y **confianza** (confidence) [32]:

- Soporte: Mide el porcentaje de veces que se produce una coincidencia entre los datos a analizar. Se calcula dividiendo la cantidad de veces que aparece dicha coincidencia entre el número de datos. Para hacer los cálculos necesarios se emplea la ecuación $\boldsymbol{sop(X) = \frac{|X|}{|D|}}$, siendo X el número de coincidencias y D el número de casos de prueba para montar el modelo de los que se dispone.

- Confianza: Mide la cantidad de veces que se prueba que una coincidencia es cierta. Para calcular esta medida, se utiliza el soporte de la coincidencia, incluida la solución, dividida entre el soporte de la coincidencia. Esto corresponde con la siguiente fórmula: $conf(X \Rightarrow Y) = \frac{sop(X \cup Y)}{sop(X)} = \frac{|X \cup Y|}{|X|}$, siendo X la coincidencia y Y la solución a la que se puede llegar según esa coincidencia.

Para que una regla sea aceptada, tiene que cumplir unos mínimos de soporte y confianza que marca el usuario. Para que eso sea posible, primero se pasan todas las reglas por el soporte para, en un paso posterior, aplicar la fórmula de confianza a todas aquellas que pasen el filtro del soporte mínimo.

Con todos aquellos datos que lo hacen cierto forma un hijo derecho y con los que no, forma un hijo izquierdo. Esto se repite hasta que ya no sea posible separar un nodo, en cuyo caso formaría un nodo-hoja, que coincide, además, con una de las posibles soluciones.

A la hora de realizar la predicción, el dato recorre el árbol siguiendo un camino en función de las reglas hasta llegar a un nodo-hoja. El objetivo con el corresponda el nodo será el resultado de la predicción de ese dato.

Debido a que este tipo de predicción hace un recorrido parcial del árbol, la complejidad es logarítmica en base a la cantidad de datos que haya **O(**log n**)** [33].

Dado que nuestro conjunto de estados objetivo es cualitativo, antes de realizar cualquier ejecución del algoritmo, hay que codificar los estados para darles valor numérico. Hemos decidido que los valores queden de la siguiente manera: 1 para suspenso, 2 para aprobado, 3 para notable y 4 para sobresaliente. Una vez obtenidos los resultados de la predicción, habrá que realizar la operación inversa y decodificarlos.

A continuación, se presenta un diagrama con el contenido de un árbol una vez construido el modelo con los datos de los que disponemos:

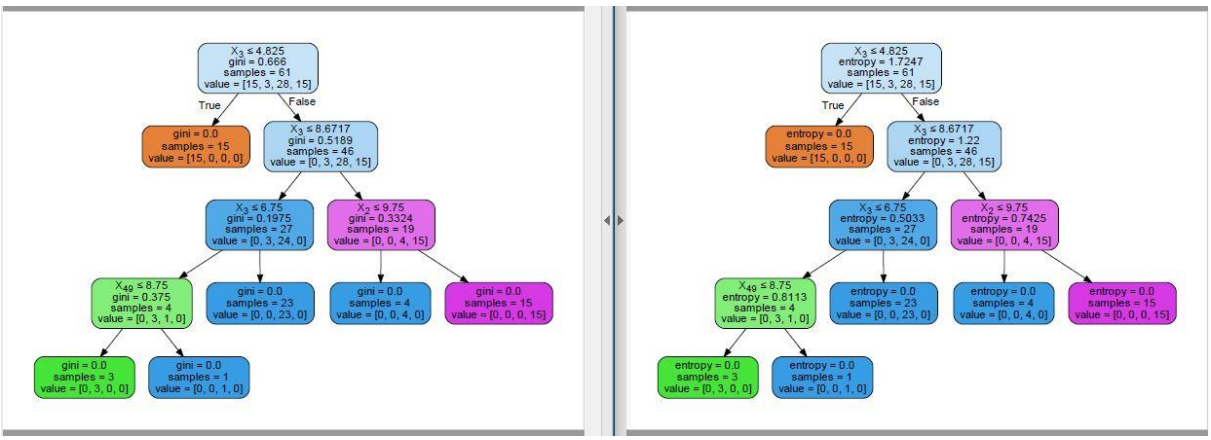


Ilustración 48: Modelo de árboles de decisión

En todos los nodos del árbol se pueden ver tres variables:

- gini [34]/entropy:
  Esta variable mide la calidad de la división de los datos. Puede hacerse bajo dos criterios: la impureza de Gini o por entropía. Tras probar ambas y no obtener diferencias de resultados, en el proyecto se utiliza la programada por defecto, es decir, la impureza de Gini; que indica la probabilidad de que un elemento se etiquete de forma incorrecta.

  Como se puede observar, a medida que se baja en el árbol desde la raíz, la probabilidad de un mal etiquetado disminuye. En el caso de los nodos-hoja, este valor está a cero porque no hay división de datos.

- samples:
  El valor contenido en esta variable muestra la cantidad de casos de estudio que se analizan para crear el nodo. El programador puede definir cuál es el máximo y el

mínimo de elementos de un nodo (parámetros *max_leaf_nodes* y *min_samples_leaf* respectivamente) y cuántos hacen falta como mínimo para que se pueda producir una división (parámetro *min_samples_split*). En el proyecto no hemos definido ningún límite, ni mínimo ni máximo.

- value:
  Esta variable contiene un array de números enteros en el que cada posición corresponde con cada una de las posibles calificaciones obtenibles por el alumno. El valor contenido en cada uno de los elementos del array es el número de casos de estudio que obtienen la nota que "simboliza" la posición.
  El resultado final de la predicción se calcula sumando los arrays de todos los nodos-hoja.

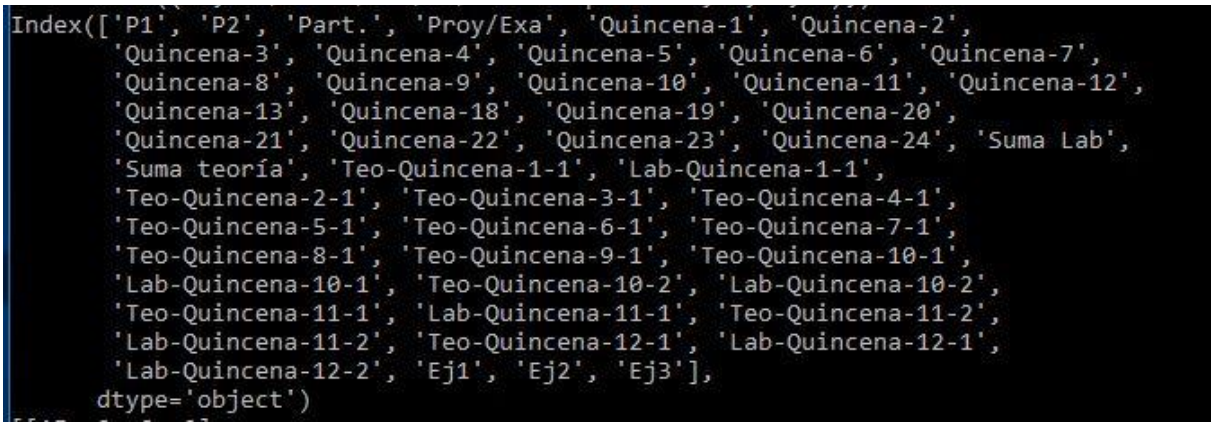


Ilustración 49: Columnas utilizadas para el árbol de decisión

Además, en los nodos-padre se encuentra un cuarto componente. Éste indica cuál de los atributos de la tupla de datos por estudiante se está teniendo en cuenta a la hora de decidir si se cumple o no (parte **Xi** de la ecuación). Además, indica también cuál es el valor límite para que se cumpla o no (parte **<= valor**). En las gráficas se puede ver que compara X_3, X_2 y X_{49} con diferentes valores. Estas variables, como se puede observar en la captura anterior, se corresponden con la nota en el proyecto o examen, la participación en la asignatura y la nota del tercer ejercicio, respectivamente.

Tomando como ejemplo el primer nodo del árbol, el componente es $X_3 \leq 4{,}825$. Esto significa que para que el nodo se haga cierto, el alumno debe sacar menos de 4,825 en su examen o proyecto. Que este primer nodo se haga cierto equivale a un suspenso en la asignatura.

7.3.2.1 Pruebas

Tras obtener el modelo, se realizaron las cuatro pruebas descritas al principio del apartado 7.1. Como se puede ver en la siguiente tabla, las tres primeras pruebas generan una tasa de error superior al 50%. Esto podría ser aceptable porque las notas de ejercicios y prácticas que se tienen en cuenta en dichas pruebas sólo suponen un 30% de la nota final en total.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Árboles de decisión	25 (52%)	27 (56’25%)	27 (56’25%)	16 (33’3%)

La última prueba, por otro lado, tiene una tasa de error del 33,3%. Esto mejora los resultados de las primeras tres pruebas.

Sin embargo, es muy común encontrarse casos como los siguientes:

Ilustración 50: Casos extremos

Como se puede observar, el margen de error entre la nota predicha y la nota real es muy pequeño. Por este motivo, para valorar si una predicción es válida o no se ha decidido tener en cuenta la Nota final real $\pm 0'25$. Con esta hipótesis, la tabla de errores queda así:

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Árboles de decisión	15 (31’25%)	15 (31’25%)	17 (35’42%)	8 (16’67%)

Teniendo en cuenta ese 0’25 de margen, la tasa de erro baja considerablemente.

7.3.3 Regresión lineal

Esta forma de predicción busca la relación entre una variable dependiente y una serie de variables independientes mediante diferentes ecuaciones, que necesitan usar un coeficiente (β) que refleja todos aquellos componentes que quedan al azar. En este caso, la variable dependiente sería la nota final. Y las independientes serían las calificaciones y comportamientos de los que ya dispone cada alumno.

En la siguiente imagen, y a modo de ejemplo, se puede observar una gráfica de una regresión lineal con una variable dependiente y dos independientes, es decir, una regresión lineal múltiple.

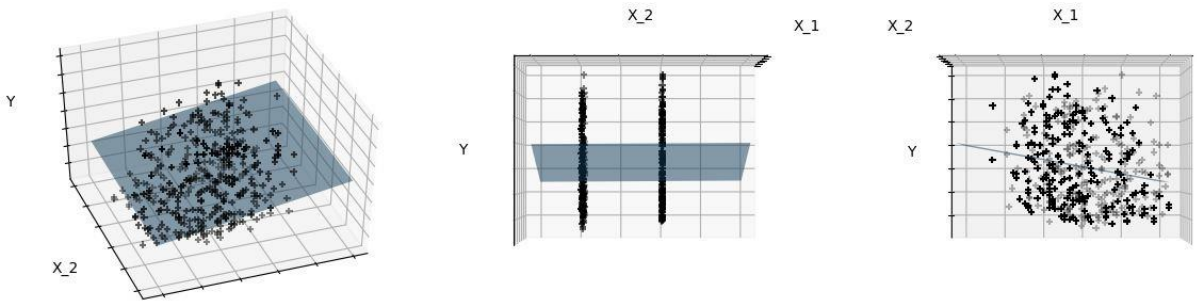


Ilustración 51: Ejemplo de regresión lineal

Como se puede observar, hay dos ejes **X** y un eje **Y**. Cada uno de los **X** corresponde con una variable independiente, y el eje de las **Y** con la variable dependiente.

Este modelo se puede expresar con la siguiente ecuación:

$$Y_i = \beta_0 + \sum \beta_i X_i + \varepsilon_i$$

donde β_i es una variable que mide la influencia que tiene la variable a la que acompaña sobre la variable a la que acompaña y $\varepsilon_i$es el error posible sobre las mediciones de cada una de las variables independientes [35].

El coeficiente β indica cómo de fuerte es la relación entre las variables independientes y dependiente, de esta forma, cuanto mayor sea (en valor absoluto) este valor, más intensa es la relación entre variables.

Al tratarse de una regresión lineal, y a diferencia de los árboles de decisión, este algoritmo no necesita un conjunto discreto de estados finales porque trabaja valores contínuos. De este modo, también debemos codificar nuestras calificaciones objetivo para darles un valor cuantitativo que permita hacer las operaciones necesarias. Estas codificaciones se encuentran entre 1 y 4, siendo 1: suspenso, 2: aprobado, 3: notable y 4: sobresaliente.

Con el modelo ya construido, se realizan las predicciones, que ofrecen resultados no enteros que se encuentran entre 0.5 y 4.5. Realizando diferentes pruebas, se llegó a la conclusión que los valores menores o iguales a 1’5 se corresponden con un suspenso, los que estén comprendidos entre 1’5 y 2’5 serán aprobados, entre 2’5 y 3’5 serán notables; y los mayores a 3’5, un sobresaliente.

7.3.3.1 Pruebas

Tras obtener el modelo, se han realizado las cuatro pruebas descritas al principio del apartado 7.1. Como se puede ver en la siguiente tabla, las dos primeras pruebas generan una tasa de error superior al 60%. Esto podría ser aceptable porque las notas de ejercicios y prácticas que se tienen en cuenta en dichas pruebas sólo suponen un 30% de la nota final.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Regresión lineal	31 (64’58%)	35 (72’92%)	20 (41’67%)	23 (47’92%)

Las dos últimas pruebas, por otro lado, tienen una tasa de error superior al 40%. Esto mejora los resultados de las primeras dos pruebas.

Sin embargo, es muy común encontrarse casos como los siguientes:

Ilustración 52: Casos extremos 2

Como se puede observar, el margen de error entre la nota predicha y la nota real es muy pequeño. Por este motivo, para valorar si una predicción es válida o no se ha decidido tener en cuenta la Nota final real $\pm 0'25$. Con esta hipótesis, la tabla de errores queda así:

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Árboles de decisión	20 (41’67%)	22 (45’83%)	18 (37’5%)	16 (33’3%)

Esto hace que la tasa de error baje considerablemente en todos los casos.

En el Apéndice D se encuentran los resultados reales completos, en los que se podrá ver, que aún con la tasa de error reducida, hay varios errores que son aceptables.

Capítulo 8: Conclusiones y Trabajo futuro

Capítulo 8: Conclusiones y trabajo futuro

8.1 Conclusiones

Tras la investigación sobre diferentes estudios y herramientas de predicción de rendimiento académico, hemos podido observar que hay un gran interés en conocer de antemano cuál puede ser el rendimiento académico de un estudiante. Sin embargo, también hemos visto que hay pocas herramientas que saquen dicho interés del marco teórico. También hemos observado la gran cantidad de información que un profesor puede llegar a tener sobre un alumno. Sin embargo, no es fácil juntarla toda para poder realizar análisis combinado de esos datos. Debido a esto, hemos decidido desarrollar una herramienta que permita al profesor visualizar la información sobre su grupo de alumnos de forma rápida y sencilla (estadísticas), sacar patrones de comportamiento (clustering) y predecir cuáles van a ser las notas finales de sus estudiantes (predicción).

Para integrar estas tres funcionalidades y poder mantener los datos lo más actualizados posible, implementamos un entorno web utilizando el microframework Flask, que facilita la programación web a la hora de comunicar HTML y Python. Además, para proporcionar la mayor cantidad posiblede información en las estadísticas, decidimos usar gráficas dinámicas. Para esto, nos apoyamos en la librería Highcharts.

En cuanto al apartado de estadísticas, mejora considerablemente la visión general de una asignatura. Esto permite al profesor ver qué ejercicios o prácticas obtienen mejores o peores calificaciones, que podría ser de especial utilidad para profesores que no tienen experiencia previa enseñando la materia.

Por otro lado, el apartado de clustering, o agrupamiento, permite juntar alumnos según su comportamiento hacia la asignatura. Después de analizar varios algoritmos, hemos decidido utilizar dos tipos de clustering: el jerárquico, que, según nuestra implementación, muestra la nota final media de cada grupo; y el KMeans, que permite ver las características habituales de cada grupo. Además, hemos podido observar que hay una relación entre la asistencia a clase y las calificaciones obtenidas en los ejercicios, y otra entre las notas de los proyectos y la cantidad de veces que un alumno interactúa con el campus virtual.

Tras analizar dos algoritmos de predicción, árboles de decisión y regresión lineal; y los resultados obtenidos en las pruebas, hemos podido observar que se comportan bastante bien en este tipo de situaciones cuando ya contamos con una gran cantidad de información. Por otro lado, sin tener en cuenta un margen de error de 0’25 puntos respecto a la nota final, en las primeras pruebas la regresión lineal tiene una tasa de error superior al 64% frente al 56% de máximo obtenido por los árboles de decisión. Es por esto que, en caso de disponer de poca información, desaconsejamos el uso de la regresión lineal.

8.2 Trabajo futuro

A continuación, se presenta una lista de posibles mejoras que se pueden implementar en un futuro con objeto de ampliar el sistema.

8.2.1 Lógica de la aplicación:

- Funcionalidad dedicada al alumno. Siguiendo la misma estructura que ha sido implementada para el profesor. De esta forma se podrían permitir funciones tales como:

    - Situar al alumno en las gráficas estadísticas con un punto. De esta forma el alumno puede ver cómo es su rendimiento en función del rendimiento global del grupo.

    - Situar al alumno en las gráficas de clustering. Esto permitiría que el alumno viese su rendimiento en función del histórico de toda la asignatura.

    - Mostrar al alumno su predicción de la nota esperada por el sistema.

- Introducir nuevos algoritmos de clustering para permitir un mayor rango de posibilidades

- Introducir nuevos algoritmos de predicción

- Realizar un comparador de algoritmos de predicción. Esto permitiría analizar qué algoritmos se comportan mejor en este sector.

- Mejorar el predictor permitiendo al profesor o al alumno hacer una predicción en función de posibles acciones. Por ejemplo, permitiéndoles especificar que un mes concreto no va a asistir. De esta forma, el algoritmo realizaría de nuevo la predicción como si no hubiera asistido en ese mes.

- Mostrar avisos a los usuarios del sistema. Por ejemplo, mandar un aviso a un alumno cuando esté cerca del suspenso. Para esta funcionalidad deberían de implementarse los cambios correspondientes en la base de datos.

- Funcionalidad dedicada al administrador: darle la posibilidad de eliminar usuarios, asignaturas o grupos y modificar los datos ya presentes en la base de datos.

- Permitir a cualquier usuario la posibilidad de cambiar su contraseña.

- Generalizar el sistema para que permita más asignaturas. Esta tarea conlleva cambios significativos en toda la aplicación. Algunas de las subtareas que deben realizarse son:

✓ Investigar el modelo de evaluación de las asignaturas que se deseen incorporar.

✓ Agregar el procesamiento de los nuevos ficheros que se necesiten, así como el tratamiento correspondiente en la base de datos.

✓ Generar su correspondiente visualización estadística.

✓ Realizar la algoritmia de preparación de los datos para la predicción, así como del clustering.

8.2.2 Cargador:

- Ampliar el soporte a más tipos de formato de los ficheros de entrada. De esta forma se podría ampliar el soporte a otras asignaturas.
- Mejora de rendimiento en el proceso de extracción.
- Mejorar el soporte a carga paralela, es decir, el proceso de carga en el caso de que varios usuarios estén realizando dicho proceso al mismo tiempo.

8.2.3 Interfaz web:

- Añadir soporte “drag and drop” a la interfaz del cargador.
- Mejorar la presentación de los formularios utilizados para los filtros de fechas en las gráficas.
- Realizar la interfaz web del alumno entera.

Capítulo 9: Conclusions and Future Work

Capítulo 9: Conclusions and Future Work

9.1 Conclusions

After doing research about different studies and tools for the prediction of academic performance, we have observed that there is great intererest in knowing in advance the possible performace of a student. However, there are not many tools to bring this interest from theory to practice. We have noticed that a teacher can have a lot of information about a student, but it is not easy put together all this info to carry out a combined analysis of al this data. Because of this, we decided to develop a tool that allows the teacher to visualize the information they have about their students in a quick and easy way (statistics), get behavioral patterns (clustering) and predict the final marks of the students (prediction).

To integrate these three functionalities and keep the data as updated as possible, we introduce a web environment using the Flask microframework. This makes web programming easier regarding the communication of HTML and Python. Besides, to include as much information as possible in the statistics, we decided to use dynamic graphs, with the help of the Highcharts library.

The statistics section significantly improves the gerenal overview of a subject. This allows the teacher to see what exercises or hand-ons get better or worse results. This could be specially useful for teachers without previous experience in the subject matter.

On the other hand, the clustering subject allows to group together students according to their behavior. After analyzing several algorithms, we decided to use two types of clustering: hierarchical, that shows the average final grade of each group, and Kmeans, that shows the recurring characteristics of each group. Moreover, we could observe that there is a relation between class assistance and grades in the exercises, and a relation between the marks in the projects and the number of occasions an student interacts with the virtual campus.

After analyzing two prediction algorithms: decision trees, and lineal regression; and the results of the tests, we observed that both ones behave really well in the kind of situations in which we have a great deal of information. On the other hand, irrespective of an error margin of 0,25 points regarding the final mark, in the first tests the linear regression has an error rate of more than 64% and the decission tree has an error rate of 56%. That is the reason why, in case of limited information available, we don’t recomend using lineal regression.

9.2 Future work

List of possible improvements that can be implemented in the future in order to expand the system:

10.2.1 Logic of the application:

- Functionality dedicated to the student following the same structure that has been implemented for the teacher, such as:
    - Place the student with a dot in the statistical graph. This way, the student can see their performance in terms of the overall performance of the group.
    - Placing the student in the clustering graphs. This way, the student can see their performance in terms of a historical of the subject.
    - Showing the student their predicted mark.
- Introducing new clustering algorithms to allow a greater range of possibilities.
- Introducing new prediction algorithms.
- Performing a predictive algorithm comparison. This would allow us to analyze which algorithm behaves better in this section.
- Improving the predictor, allowing the teacher or the student to make a prediction based on different scenarios. For example, allowing them to specify that they are not going to attend class a given month and performing the prediction again.
- Showing warnings to the users of the system. For example, send a notice to a student when they are close to failing. For this function, the corresponding changes in the database should be implemented.
- Functionality for the administrator: Giving them the posibility to delete users, subjects or groups and modify the data already present in the database.
- Allowing any user to change their password.
- Making it possible for the system to include more subjects. This task implies significative changes in all the application. These are some of the sub-tasks to carry out:
    - Investigate the evaluation model of the subjects to include.

- Add up the process for the new necessary files, as well as the corresponding treatment of the database. Generate the corresponding statistical visualization.

- Carry out the algoritmic preparation of the data for the prediction and clustering.

9.2.2 Loader:

- Extend support for more types of input files. This way you could extend the support to other subjects.
- Performance inprovement in the extraction process.
- Improve support for parallel loading, ie the loading process in case of several users doing this process at the same time.

9.2.3 Web interface:

- Add “drag and drop” support to the loader interface.
- Improve the appearance of the forms used for the date filters in the graphs.
- Develop the whole interface for the student.

Capítulo 10: Aportaciones individuales al proyecto

Capítulo 10: Aportaciones individuales al proyecto

En este capítulo se presenta el trabajo individual que ha realizado cada integrante del grupo, así como los problemas que se han presentado en el proceso.

Marco Antonio Cuevas Redondo

En las primeras fases de desarrollo, se tuvo que dedicar tiempo y esfuerzo en la investigación de herramientas y lenguajes. Todo esto con el objetivo de buscar aquellas herramientas o lenguajes que mejor se ajustaran a nuestro proyecto.

En dicha fase, este integrante del grupo se encargó de hacer comparativas sobre los lenguajes R y Python. En este punto se dedicó, sobre todo, a realizar numerosas pruebas sobre las librerías Numpy y Pandas. Finalmente, R quedó descartado del proyecto, dejándonos Python como lenguaje base.

Para hacer la comparación entre los dos lenguajes mencionados, se trataron puntos tales como:

- Calidad de las librerías de análisis.
- Soporte de la comunidad.
- Sencillez y potencia de uso.

Todo esto puede verse en el **Apéndice B** con más detalle.

En cuanto a la librería Pandas, que acabamos de mencionar, realizó un estudio más en profundidad soportándose en manuales, foros y webs que hablaban de la librería. Todo esto debido a la importancia de esta librería en el presente proyecto.

Tras esto, este integrante del grupo se encargó de buscar formas de integrar Python en un entorno web. Se encontraron distintos entornos o sistemas que permitían esto de foma sencilla. No obstante, se enfocó en el estudio de un microframework llamado FLASK y un framework llamado Django. Para esto, se dedicó a hacer pruebas básicas con el microframework Flask y a comparar dichas pruebas con el framework DJango [36]. Todas estas pruebas giraron en torno a varios puntos. Entre ellos tenemos:

- Facilidad de uso.
- Soporte de librerías o funcionalidades agragadas.

No obstante, Django fue finalmente desechado ya que Flask nos proporcionaba mayor grado de flexibilidad [37] [38].

Más adelante, comenzamos a realizar una fase de análisis de los datos que poseíamos acerca de los estudiantes. Para esta fase, este integrante se dedicó a realizar la visualización de las estadísticas a través de la librería matplotlib. Además, tomó parte en el análisis de las gráficas obtenidas y de la toma de decisiones. Para ello, implementó métodos genéricos que mostraban información estadística relevante para, posteriormente, proceder a analizarla.

Una vez analizada la información, procedió, junto a su compañera, a hacer un análisis de requisitos en cuanto a la funcionalidad estadística y de visualización. Tras esto, comenzó a diseñar e implementar la interfaz web.

En lo que respecta a la interfaz web, comenzó realizando un prototipo interactivo de la web a través de un software de prototipado rápido llamado “MyBalsamiq”. Se encargó de la creación de gran parte de las páginas que se pueden encontrar y realizó la totalidad del diseño de la web a través de las hojas de estilo CSS.

Para continuar, se dedicó a la gestión de usuarios, así como la gestión de sesiones dentro de la web. Para ello, comenzó a investigar librerías para Flask que dotasen a este microframework de dicha funcionalidad. La librería elegida fue Flask-Login, la cual permite poner etiquetas de seguridad a las URL así como guardar la sesión a través de cookies en el navegador de forma sencilla.

En cuanto a la base de datos, tomó parte en el diseño de entidades de la misma así como en su implementación. Además de esto, fue el encargado de realizar, en su totalidad, la capa de acceso a base de datos. Para ello, realizó una serie de funciones genéricas.

Posteriormente, se dedicó a la implementación del controlador con objeto de dar soporte a todas las posibles peticiones necesarias para la lógica de la aplicación.

A continuación, hablamos de la lógica de la aplicación:

- En lo referente a la parte de generación de estadísticas, tomó parte en el proceso de toma de decisiones, así como ayudar en el proceso de desarrollo, dando soporte de visualización a través de la interfaz web.

- Pasando a hablar del proceso de clustering, comenzó con la investigación del proceso de clustering jerárquico.

  Puesto que la representación a partir de un dendrograma no resultaba intuitiva a usuarios que no estuviesen familiarizados con el clustering jerárquico, diseñó un sistema de visualización alternativo. Para ello creó el sistema de representación en forma de gráfica circular que se puede en su correspondiente apartado de este documento.

  Además de esto, realizó la totalidad de la algoritmia dedicada a este apartado de clustering jerárquico.

Para finalizar, este integrante participó a partes iguales con el resto de participantes para la generación de la presente memoria.

Marta Estévez Bravo

La primera fase del proyecto fue investigar y decidir el lenguaje en el que se iba a desarrollar la aplicación. Las opciones valoradas fueron Python y R, para, finalmente, quedarnos con Python. Aunque el mayor peso del análisis de los lenguajes lo llevó el otro miembro del equipo, este componente participó en la toma de decisión sobre que lenguaje utilizar.

Después se hizo una investigación sobre los estudios, trabajos previos y herramientas ya existentes relacionados con el proyecto. Nuestra búsqueda se centró, básicamente, en localizar estudios relacionados con predicción y los métodos más adecuados para ello. También encontramos herramientas que muestran a los profesores gráficas con la información de sus alumnos. Este miembro del equipo se dedicó a profundizar en este tema y buscar información sobre la cuál empezar.

Acto seguido, empezamos con el análisis y comprensión de los datos de los que íbamos a disponer, como cargarlos y analizarlos. Para facilitar el trabajo, hicimos una implementación del ciclo de datos, primero de forma básica, cargando los archivos y mostrando algunas gráficas sencillas con Matplotlib. Este componente ha participado en esta fase activamente.

A continuación, procedimos a buscar relaciones entre datos y cómo se podían combinar. Este componente se ha dedicado entender los datos y cómo se relacionan; y a conocer algoritmos de clustering, especialmente los no jerárquicos. Las características que se buscaron son:

- Facilidad de interpretar los resultados obtenidos por un usuario no experto.

- Capacidad de utilizarlo para encontrar patrones de comportamiento.

Después de ver varios algoritmos, decidimos usar el algoritmo KMeans, con una implementación que devuelve los centroides de cada grupo. De esta parte se encargó este componente casi en su totalidad.

La siguiente fase consistió en definir las estructuras del directorio de almacenamiento de datos, de la web y de la base de datos. De esto nos encargamos a partes iguales los dos componentes del equipo.

Con la estructura web definida, empezamos una fase de prototipado de la herramienta. Para esto, utilizamos la herramienta “myBalsamiq”. Diseñamos la interfaz de las distintas pantallas y cómo se integran las distintas funcionalidades, pera después unirlas y hacer una simulación del funcionamiento. Este componente colaboró con el desarrollo de parte de los mockups.

Teniendo un diseño inicial y los algoritmos elegidos, empezamos con la implementación. Este componente colaboró en la creación de algunas de las páginas web, sin contar con el CSS, y se encargó de la integración total de las gráficas Highcharts en la parte web. Además, este componente se encargó de:

- Estadísticas: implementación de los métodos genéricos para calcular medias, varianzas y desviaciones típicas.

- Clustering: adaptación de datos para usarlos con el algoritmo KMeans, preparación de los resultados obtenidos para representarlos gráficamente. Para esta funcionalidad hemos elegido dos tipos de gráficas diferentes: lineal y poligonal.

- Funcionalidad del administrador: creación de los formularios necesarios para dar de alta usuarios, asignaturas, grupos e incluir alumnos en un determinado grupo. Además implementó los métodos específicos para añadir la información introducida en dichos formularios en la base de datos.

Por último, nos pusimos a investigar, estudiar y probar distintos algoritmos de predicción. Para esta funcionalidad nos decidimos por usar técnicas de aprendizaje supervisado. Una vez probados los algoritmos de árboles de árboles de decisión y de regresión lineal, empezamos a incluirlos en el proyecto. El primero que introdujimos fue árboles de decisión. Hicimos las pruebas necesarias y, una vez que conseguimos que funcionara, incluimos regresión lineal. De eso se encargó este componente del equipo. El proceso que siguió para implementar esta funcionalidad es el siguiente:

- Implementar los métodos necesarios para la preparación de los datos de entrenamiento y de test para pasárselos al algoritmo.

- Implementar un método que transforma las salidas esperadas reales en un conjunto finito de números enteros.

- Aplicación del algoritmo de predicción elegido.

- Analizar la tasa de acierto que proporciona el algoritmo con los datos de test y calcular su matriz de confusión.

- Adaptación de los resultados obtenidos para poder mostrarlos en el entorno web.

- Creación de la página que va a contener los resultados de la predicción en la aplicación web.

- Creación de un filtro para que el usuario pueda acotar la franja temporal de los datos que quiere utilizar para hacer la predicción.

- Hacer pruebas, preparar los datos de un grupo diferente al utilizado para crear el modelo y predecir sus notas finales.

A lo largo del proyecto hemos hecho pruebas y corregido errores en todo lo mencionado anteriormente. En esto hemos participado a partes iguales los dos miembros del grupo.

De la misma manera, la presente memoria la hemos escrito los dos de forma equitativa.

Apéndice A: Manual de instalación

Este manual corresponde a la instalación de la herramienta en un entorno Windows. Para la correcta instalación y ejecución de esta herramienta, se deben llevar a cabo una serie de pasos:

- Instalar MySQLWorkbench o XAMPP e importar la base de datos al SGBD
- Instalar el intérprete de Python
- Instalar todas las librerías necesarias de Python
- Ejecutar el sistema

11.1.1 Instalar MySQLWorkbench o XAMPP e importar la base de datos

Para comenzar, empezaremos explicando la instalación de MySQLWorkbench. Procedemos a descargarlo de su sitio oficial: https://dev.mysql.com/downloads/workbench/.

Una vez descargado, procedemos con los pasos de instalación. Nos aparecerá un instalador básico en el cuál tendremos que seguir los pasos correspondientes:

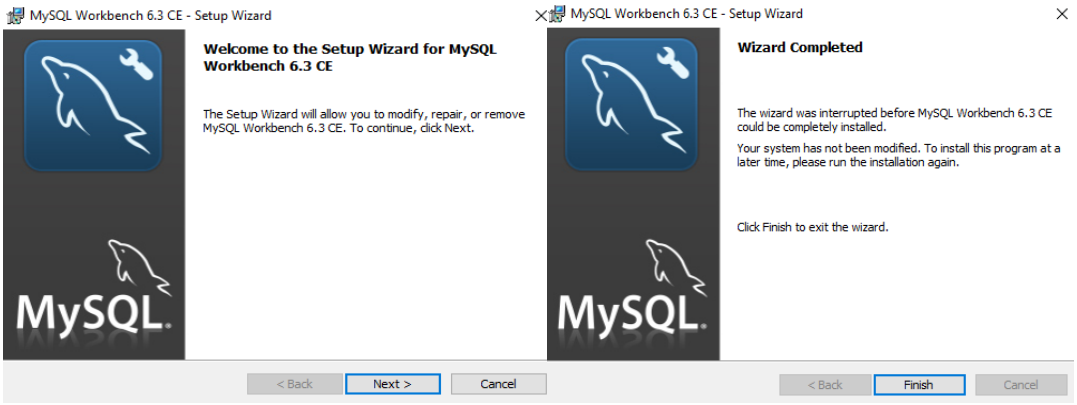


Ilustración 53: Instalación de MySQL Workbench

Una vez MYSQLWorkbench ha sido instalado, lo iniciamos y nos aparecerá el SGBD. Procederemos a asignar una contraseña para el administrador del sistema y después haremos doble click en la única conexión de MySQL que tenemos por defecto. Esto se puede apreciar en la siguiente imagen:

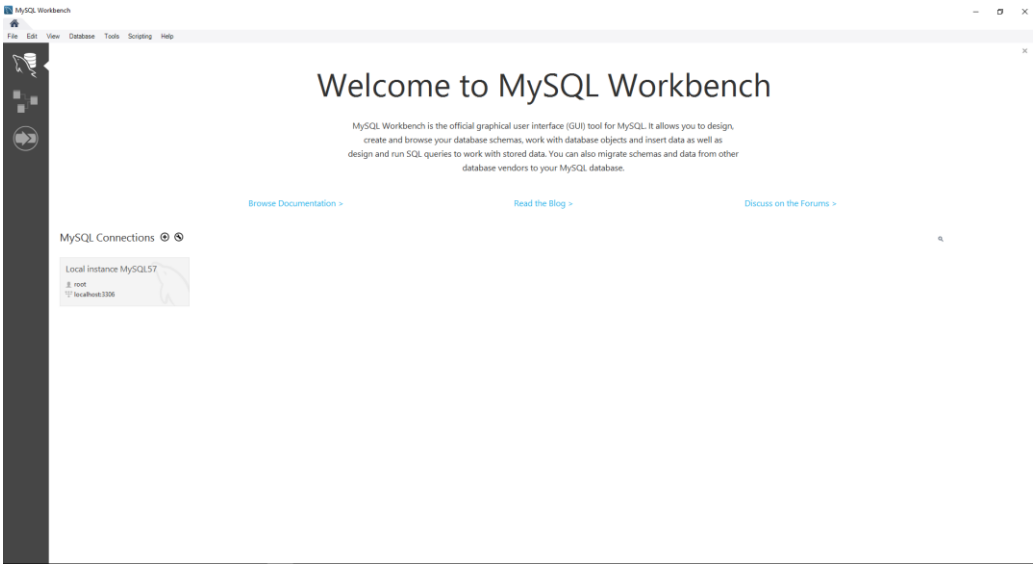


Ilustración 54: Inicio de MySQL Workbench

A continuación, deberemos ir a ***"Startup / Shutdown"*** para conectar el servidor. Le damos al botón *"Start server"* e introducimos la contraseña en el cuadro que nos aparecerá a continuación.

Después, vamos a la sección **"Schemas"** y hacemos doble click. Pulsamos la opción "Create new schema". Introducimos el nombre de la base de datos **"parend"** y aceptamos.

Una vez creada esta base de datos vacía, nos vamos a "Data Import/Restore" y seleccionamos, en la ruta del fichero, el script SQL que se puede encontrar en el proyecto. Además, seleccionaremos como "Default Target Schema" la base de datos vacía que acabamos de crear.

Una vez se han realizado estos pasos, ya tendremos la base de datos lista para funcionar. En caso de querer ejecutarlo únicamente tendremos que asegurarnos de que el servidor está conectado.

Importante: Si elegimos utilizar este SGBD, deberemos incluir la contraseña que hayamos elegido para el mismo en el fichero **"dao.py"**. De esta forma, deberemos especificar la variable **PASS** con la contraseña elegida en el SGBD.

```python
import MySQLdb

################################  Propiedades  ###################

HOST_NAME = 'localhost'
USER_NAME = 'root'
PASS = 'PASSWORD'
DATABASE_NAME = 'parend'


#########################  Ejecución de sentencia genérica  #####

def ejecutar_sentencia(sentencia):
```

Ilustración 55: Configuración del cliente de base de datos

Por otro lado, para instalar XAMPP empezamos por descargarlo de su página oficial (https://www.apachefriends.org/es/download.html).

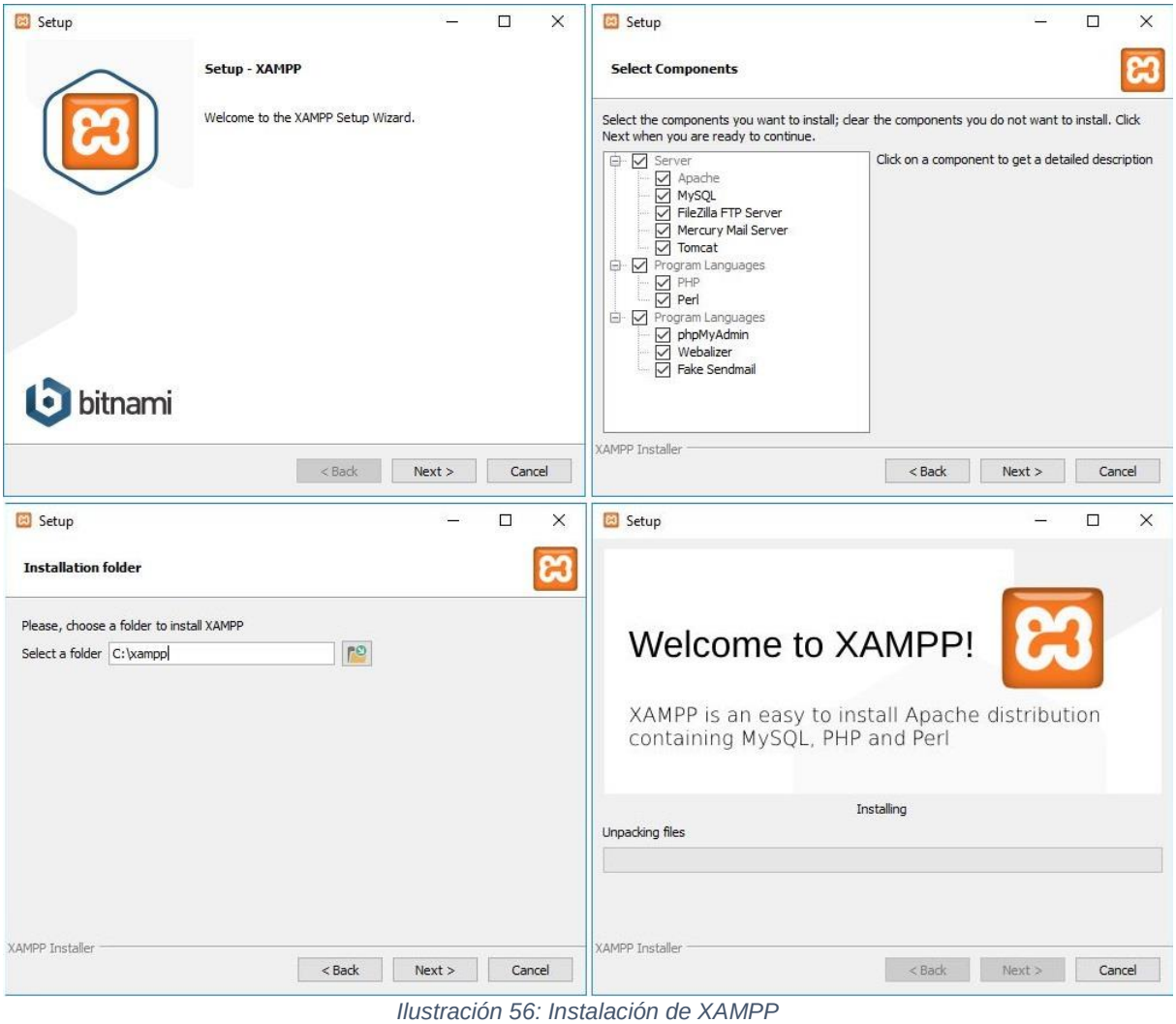


Ilustración 56: Instalación de XAMPP

Una vez lo tenemos, ejecutamos el instalador y seguimos los pasos que indica. Cuando termine, ya puede utilizarse XAMPP.

Para iniciarlo, no hay más que buscarlo en la lista de programas en el botón de inicio de Windows o como un acceso directo en el escritorio y hacer click en él. Hecho esto, saldrá el panel de control:

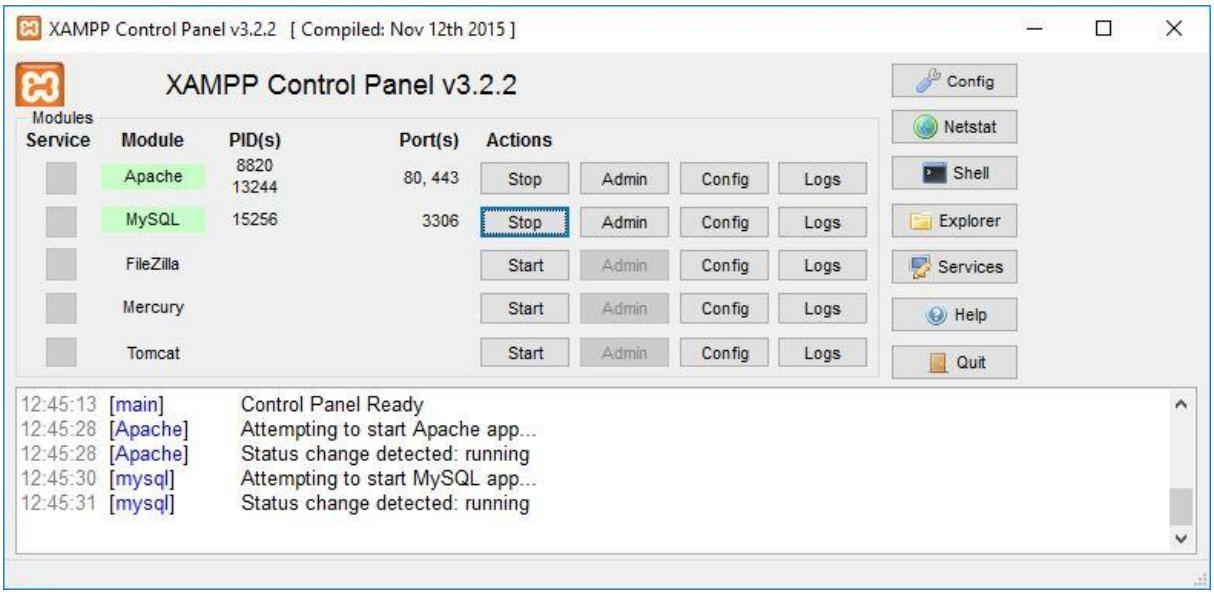


Ilustración 57: Panel de control de XAMPP

Nada más abrirse, aparece totalmente en gris. Para levantar el servidor de MySQL y así poder usar las bases de datos, hay que pulsar los botones de **"Start"** que están al lado de **Apache** y **MySQL** (y que en la imagen ponen **"Stop"**).

Ahora ya se puede acceder a las bases de datos. Para ello, hay que abrir un navegador y, en la barra de navegación, teclear ***http://localhost/phpmyadmin/***

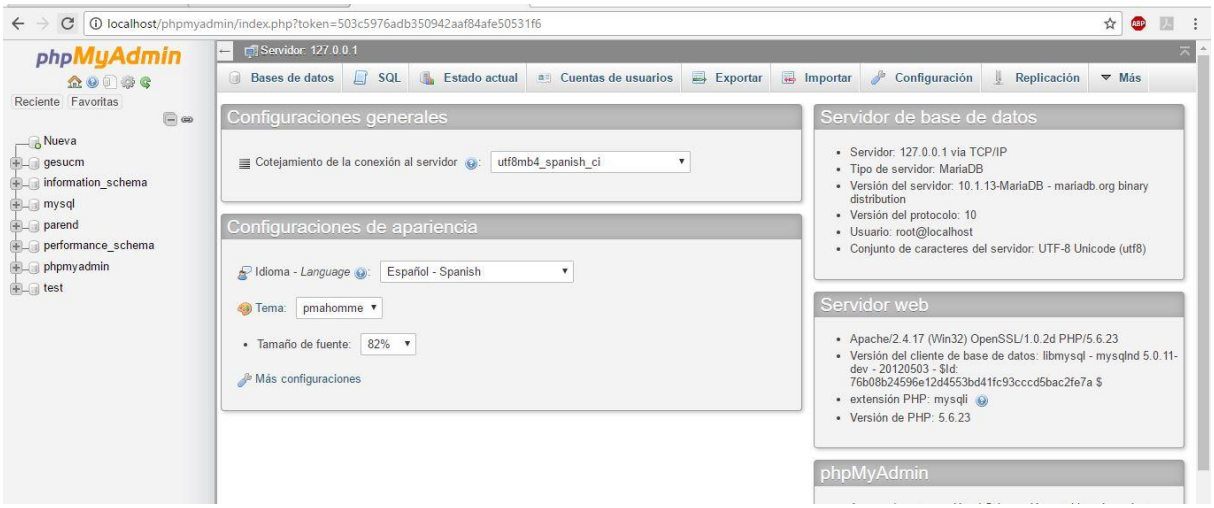
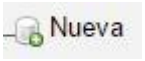


Ilustración 58: Vista principal de phpMyAdmin

Para importar una base de datos, se pulsa 

y se le pone el nombre **"parend"** en el formulario que sale por pantalla. Acto seguido, se pulsa 

. Saldrá una pantalla como está:

Ilustración 59: Importación de la base de datos en phpMyAdmin

Se pulsa en el botón que pone "Seleccionar archivo" y se elige el archivo en el que está el código y la información contenida en la base de datos.

11.1.2 Instalar Python

Para instalar Python, únicamente vamos a página oficial (https://www.python.org/), descargamos el instalador y seguimos los pasos que nos marque.

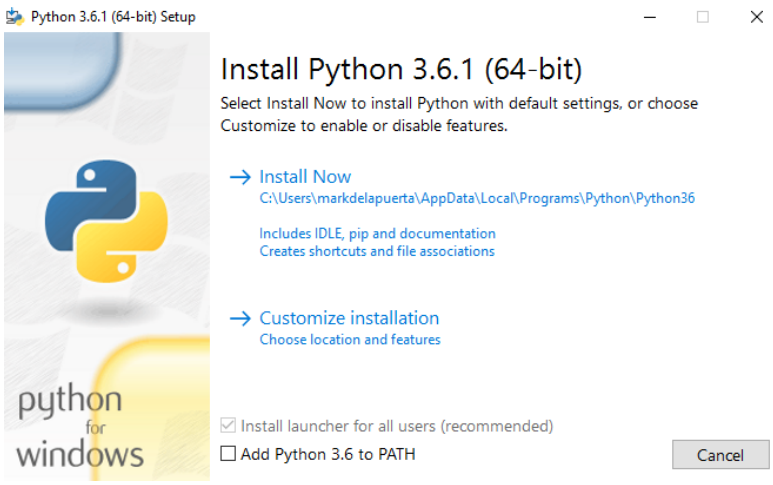


Ilustración 60: Instalación de Python

Hay que prestar especial atención al checkbox **"Add Python 3.6 to PATH"** que se encuentra en la parte inferior de la imagen. Es recomendable marcarlo para no tener que realizar el proceso manualmente en un futuro.

Una vez que Python está instalado, es necesario incluir en el sistema las librerías necesarias para que el proyecto funcione. Esto se hace en el Símbolo del Sistema (cmd) mediante el comando **pip install <librería>**.

Como primer paso, es recomendable actualizar el propio pip, utilizando el comando **python -m pip install -U pip**. Una vez hecho esto, ya se puede proceder a la instalación de las librerías: numpy, pandas, sklearn, Highchart, flask, flask_wtf, flask_login y wtforms, MySQLdb. Es necesaria la instalación de dos librerías más, scipy y una versión diferente de numpy, pero el proceso para lograrlo es ligeramente más complejo.

En primer lugar hay que descargar los archivos .whl que las contienen desde los siguientes enlaces: http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy para numpy y http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy para scipy. Después hay que situarse en la carpeta contenedora de los archivos descargados y, por último, hacer **pip install <nombre completo del archivo>**.

Importante: Instalar antes numpy que scipy.

Apéndice B: Comparativa de Python y R

A continuación, presentamos las cualidades que se han obtenido al analizar los lenguajes Python y R

11.2.1 Características de Python:

- Fácil de aprender
- Uso de librerías potentes
- Sintaxis sencilla pero eficiente
- Enfatiza en la productividad
- Comunidad muy activa que proporciona gran cantidad de paquetes
- Evoluciona muy rápido en el ámbito estadístico
- Permite una integración sencilla en entornos web con objetivo de ponerse en producción
- Gran cantidad de librerías de análisis de datos
- Gran cantidad de librerías de visualización de datos
- Facilidad a la hora de trabajar con datos a través de IPython Notebook
- Código fácilmente reutilizable

11.2.1 Características de R:

- Potente capacidad de visualizar datos
- Comunidad activa que proporciona gran cantidad de librerías de análisis
- Capacidad para realizar cómputos independientes
- Funciona con la mayoría de los tipos de datos de forma muy ligera
- Capacidad de manejo de grandes cantidades de datos (buena solución para proyectos de Big Data)

- Curva de aprendizaje elevada

Las dos listas anteriores las hemos obtenido tras el estudio del siguiente material:

- *Python Cookbook*, de David Beazley y Brian K. Jones [39].
- La web "Introducción al Procesamiento de Datos con Python" [40].
- *An Introduction to Statistical Learning with Applications in R*, de Gareth James et al. [41]
- Apuntes de la UNED (ETS Ingeniería Informática): "Introducción al análisis de datos con R", de Alfonso Urquía Moraleda y Carla Martín Villalba [42].
- Apuntes de la Universidad del País Vasco: "Lectura, manipulación y análisis de datos en R" de F. Tusell [43].
- Tutorial de R [44].
- El artículo "Análisis de datos: ¿R o Python? Infografía", de Maribel Tirados [45].

Apéndice C: Implementación del cargador

En este apartado explicaremos el proceso de carga de los datos al servidor a través de la interfaz web del sistema. Para la correcta realización del proceso de carga, se presenta al usuario un formulario en el cual debe introducir una serie de ficheros con un formato válido que se especificará más adelante.

Para esto, se proporciona un formulario al estilo de flask que posteriormente se valida en el servidor. A continuación, mostramos una imagen de su funcionamiento.

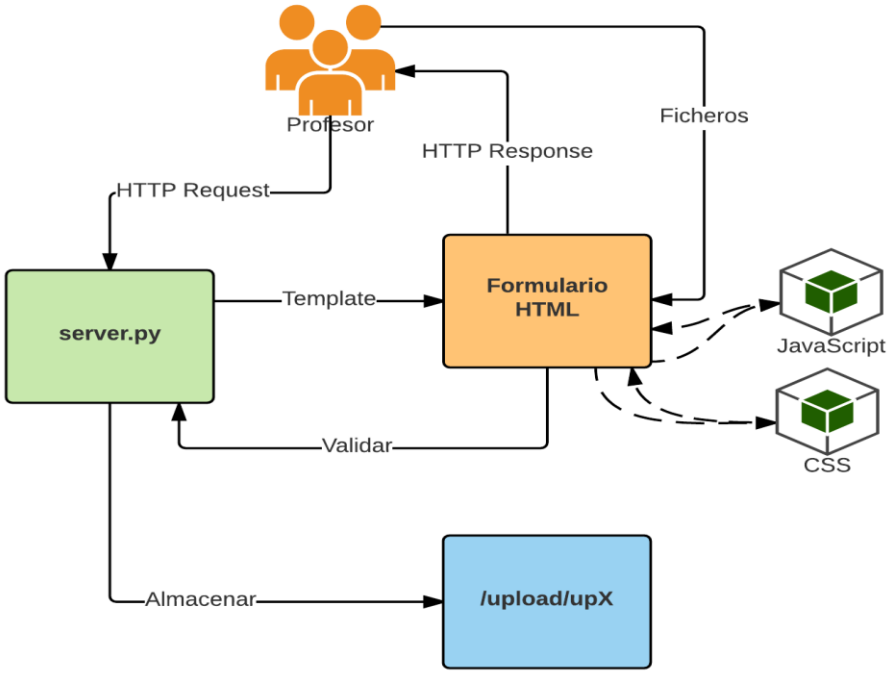


Ilustración 61: Esquema de implementación del cargador

A continuación, ponemos los pasos, en orden, que se siguen en el diagrama anterior:

- Para empezar, el profesor solicita la página donde se encuentra el formulario a través de una petición **GET**.

- El servidor procesa la petición y busca el **Template**. Este se sirve de **CSS** y **JavaScript** para completar su vista. El formulario HTML posee la línea de código "{{ form.group }}" que introduce el campo correspondiente de subida de ficheros mediante la sintaxis de flask.

- Posteriormente se renderiza el Formulario al profesor.

- El profesor envía los ficheros a través del formulario al servidor.

- El servidor almacena los ficheros en **/upload/upX**. La **X** es un número generado de tal forma que el path calculado no se esté siendo usado (no esté creado). Esto nos

permite realizar una carga paralela de datos desde dos puntos. Para ello obtenemos los datos del template a través de la petición **POST**. Se leen mediante **“request.form”** y se comprueba si estos han sido introducidos utilizando el formulario a través de **“form.validate_on_submit()”**

Apéndice D: Diseño de la interfaz web

En este apartado mostraremos algunos de los que se realizaron el proceso de diseño de la interfaz web.

En esta primera captura podemos ver el diseño de la página principal:

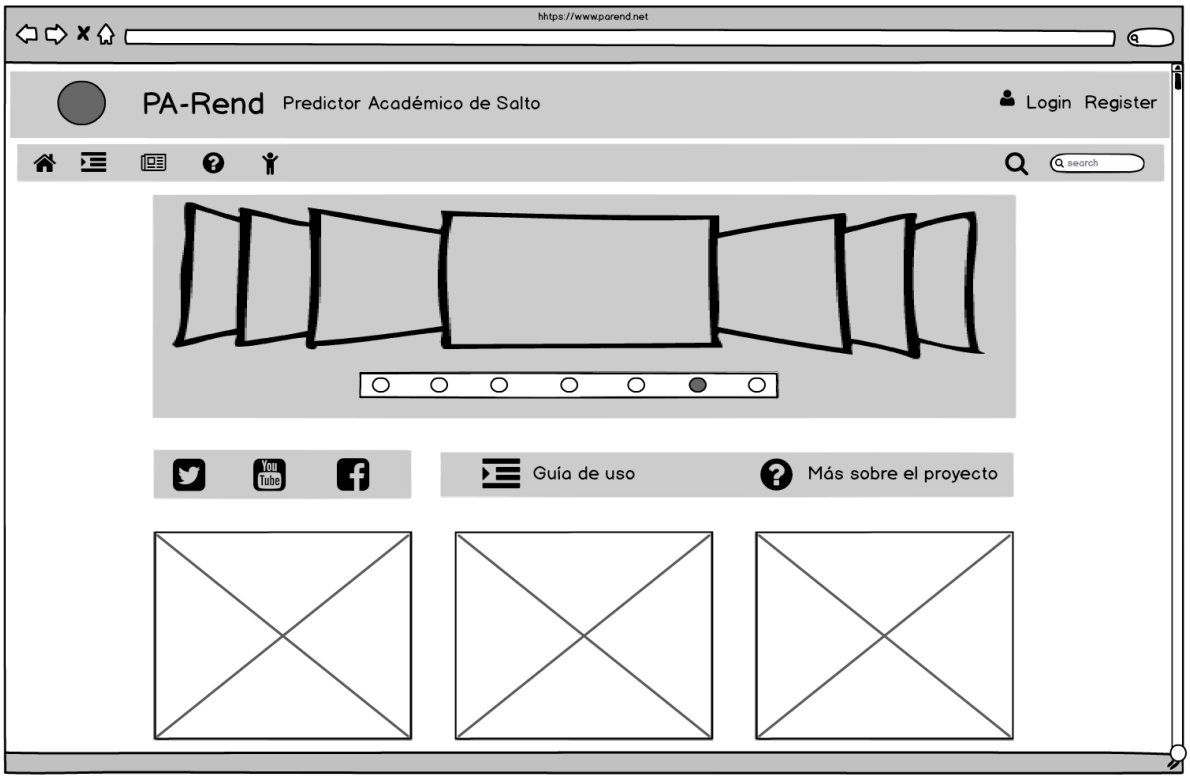


Ilustración 62: Mockup, página de inicio

Las dos siguientes imágenes muestran el diseño realizado para el login y el registro:

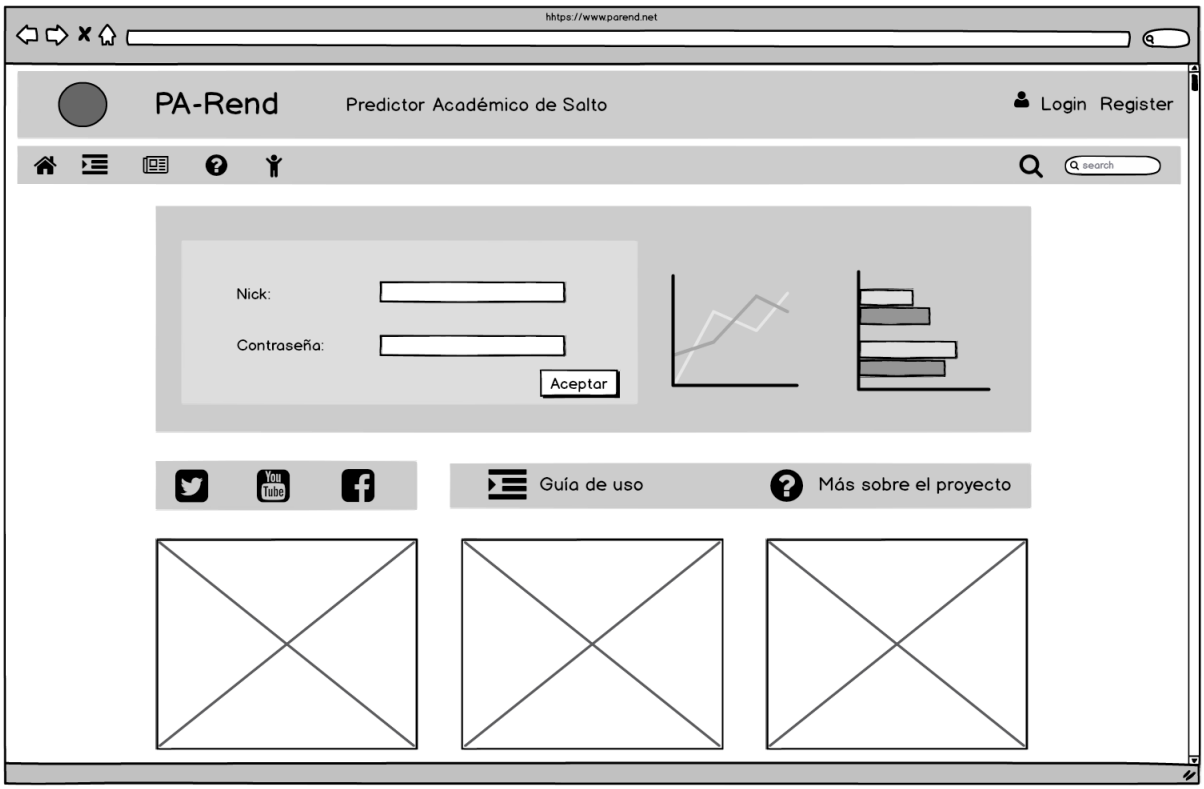


Ilustración 63: Mockup, página de login

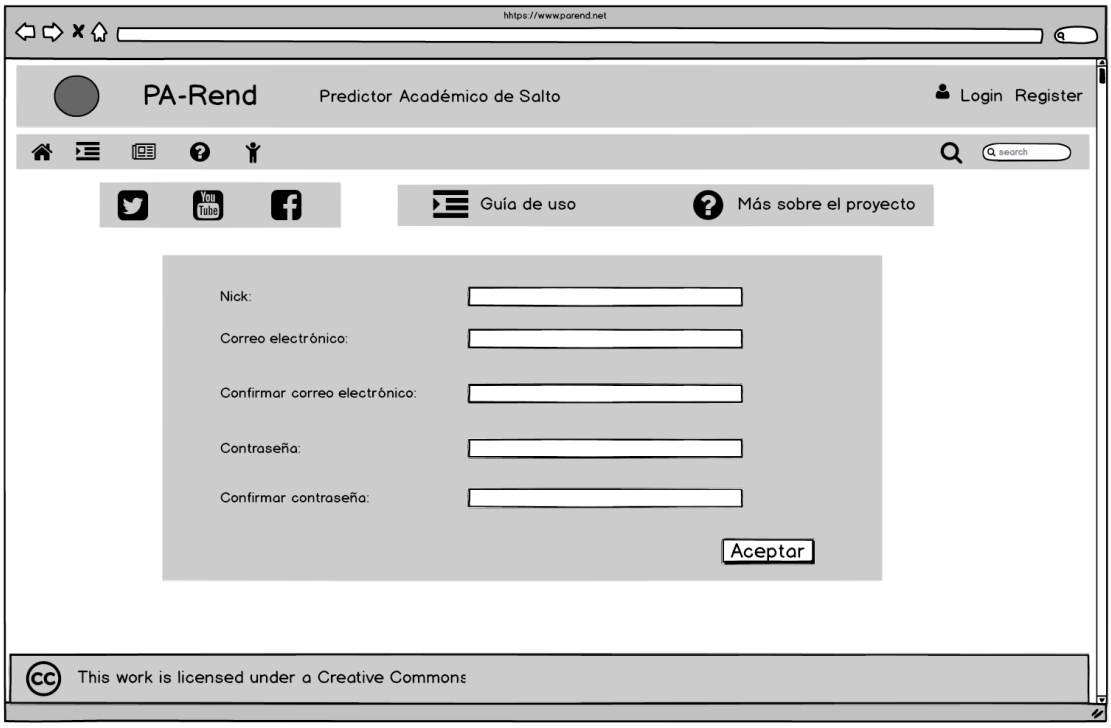


Ilustración 64: Mockup, página de registro

La siguiente imagen muestra el diseño de la interfaz del formulario web:

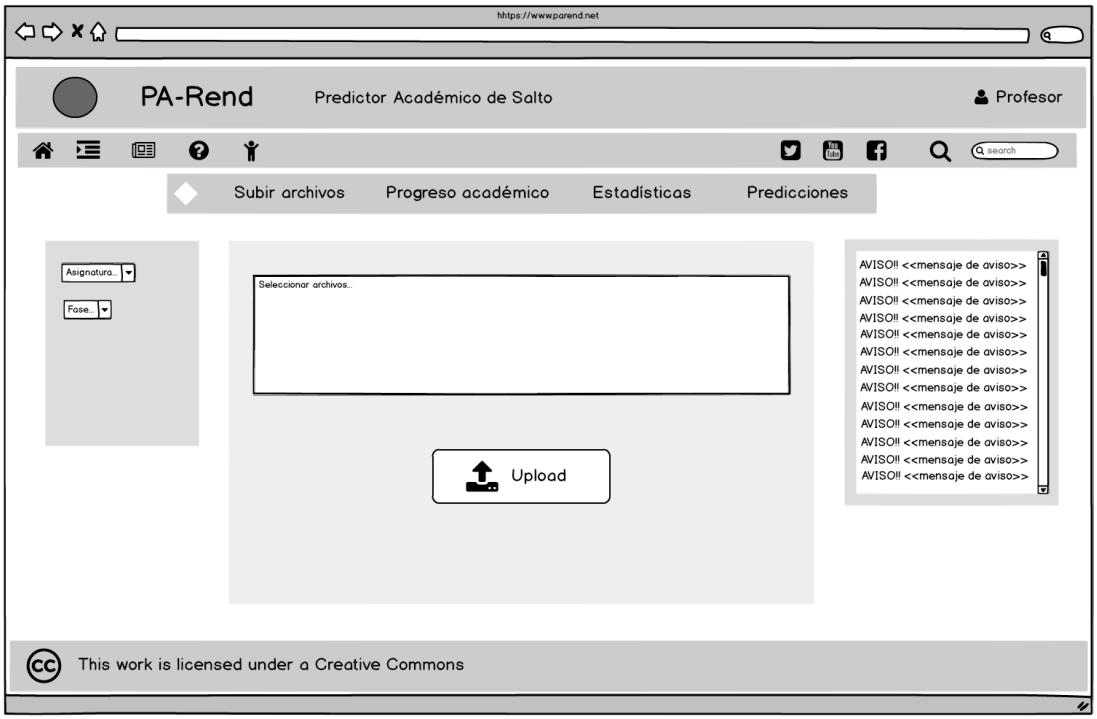


Ilustración 65: Mockup, página de carga

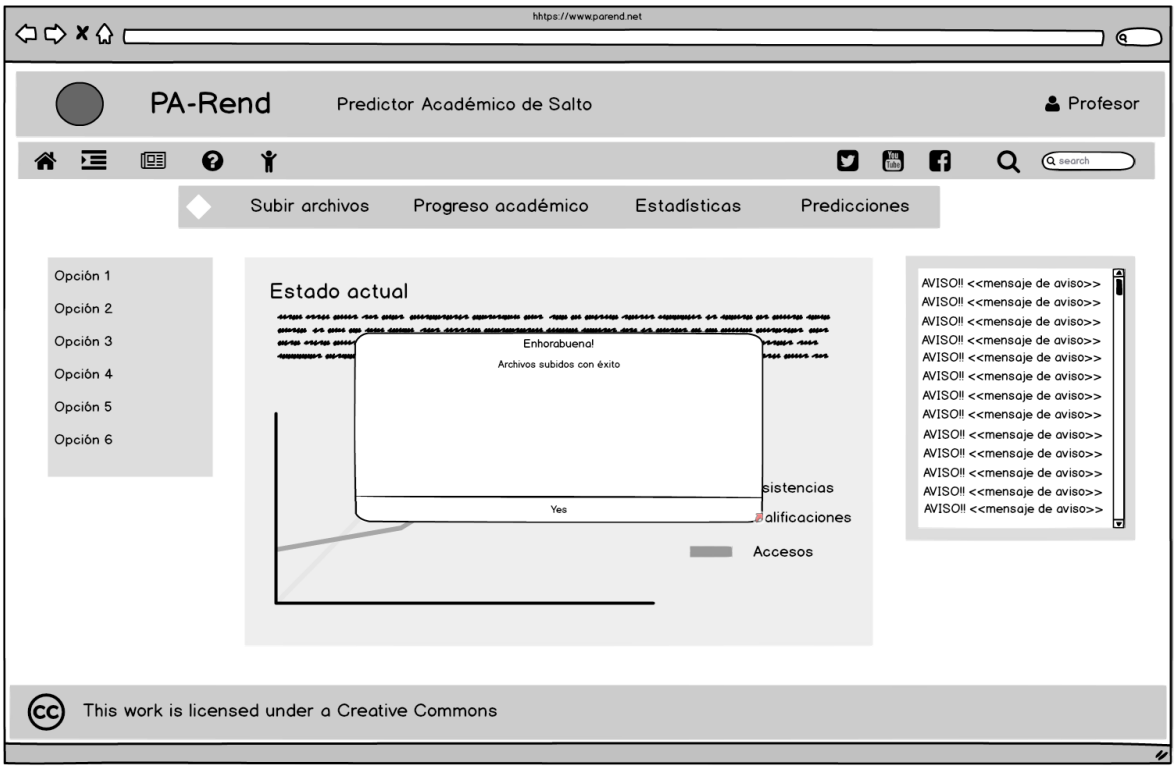


Ilustración 66: Mockup, página de carga 2

Las siguientes imágenes podemos ver los diseños básicos para las interfaces correspondientes a estadísticas, clustering y predicción:

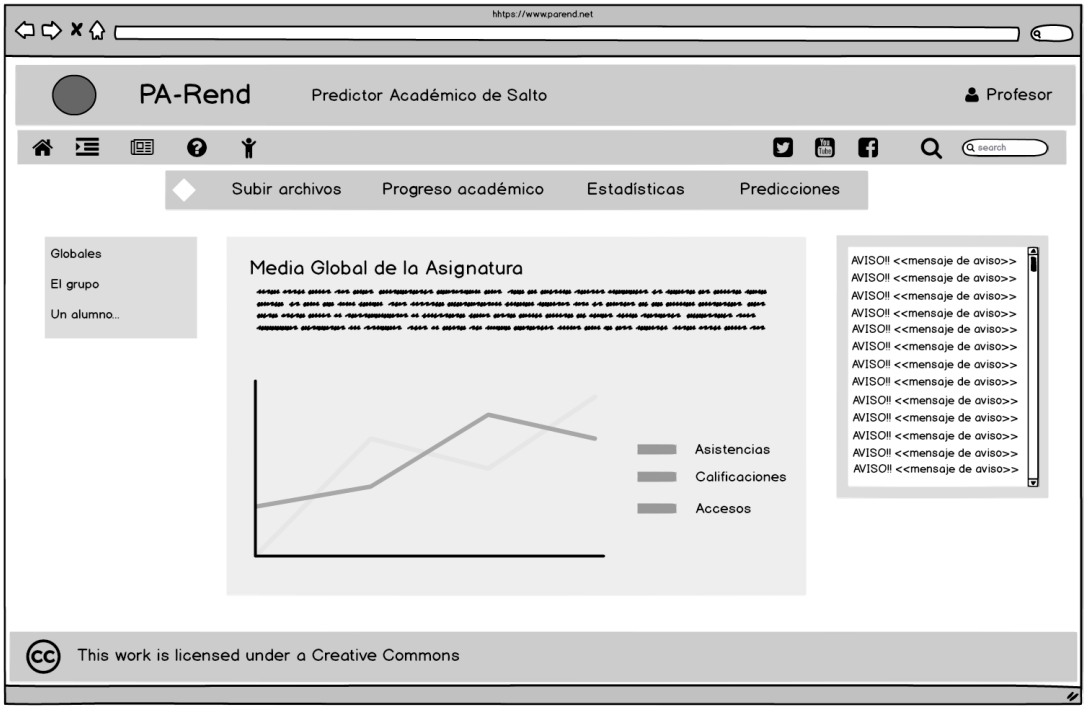


Ilustración 67: Mockup, página de estadísticas

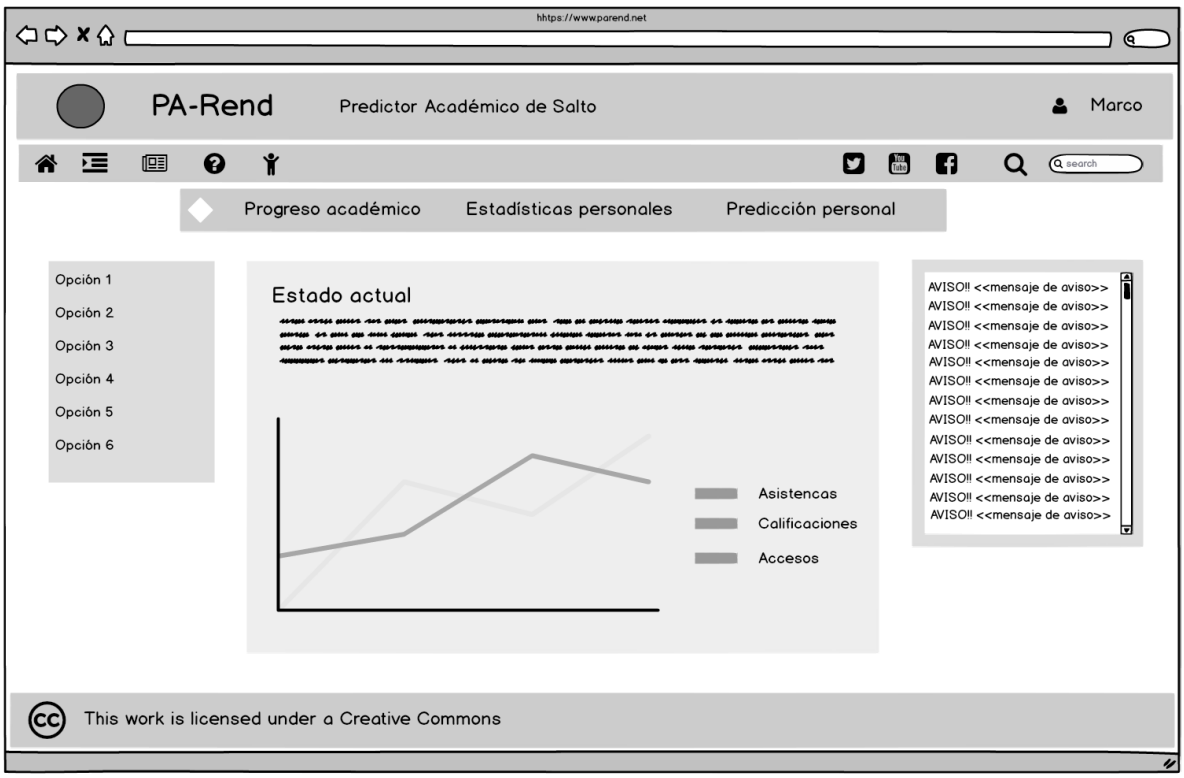


Ilustración 68: Mockup, página de clustering

Ilustración 69: Mockup, página de predicción

Apéndice E: Resultados de las pruebas

Nota: Todos los nombres que se pueden ver en estas imágenes, no son nombres reales.

- **Árbol de decisión**

Nombre	Nota real	Prueba 1	Prueba 2	Prueba 3	Prueba 4
ABDOULIE SUN	8,29	NT	NT	NT	NT
AICHA LI	8,38	NT	NT	NT	NT
ALESSANDRO DAVIS	6,57	NT	NT	NT	NT
ALEXANDER BOURAS	7,81	NT	SB	NT	NT
ALEXEY AMARA	7,59	NT	NT	NT	NT
ANDREA MILLER	6,38	NT	NT	NT	NT
ANDREY ABED	7,76	NT	NT	NT	NT
ANNE MARIE MOHAMED	8,26	NT	NT	SB	NT
ANTONIO ANDERSON	8,85	NT	NT	NT	NT
ASMA JOHNSON	9,21	NT	NT	NT	NT
BAKARY JI	9,27	NT	NT	NT	NT
CAROLINE BRAHIM	8,96	NT	NT	NT	NT
CHANTAL MOHAMED SALEM	0,00	SS	SS	SS	SS
DAIVA CLARK	8,93	NT	SB	SB	NT
DMITRY HADDAD	9,68	NT	SB	SB	NT
FATIMA CHEN	4,84	NT	NT	NT	SS
GIUSEPPE MOORE	6,83	NT	NT	SB	NT
IGOR DJILALI	7,31	SB	SB	SB	AP
INGA MARTINEZ	7,74	NT	NT	NT	NT
INGRID LARBI	6,47	NT	NT	NT	NT
IVAN BENAISSA	8,73	NT	NT	AP	NT

Nombre	Nota real	Prueba 1	Prueba 2	Prueba 3	Prueba 4
JOLANTA LEWIS	7,19	NT	NT	NT	NT
JOSEPH BROWN	6,84	NT	NT	NT	NT
KARIMA LIU	7,23	NT	NT	NT	NT
KRISTINA HARRIS	8,15	NT	NT	NT	NT
LAMIN HUANG	5,71	NT	NT	NT	AP
LAURA GARCIA	6,61	SS	SS	AP	AP
LAURA SLIMANI	8,87	SB	SB	NT	NT
LINA MARTIN	8,71	SB	NT	NT	NT
LUCA TAYLOR	8,43	NT	SB	NT	NT
MANPREET SMITH	7,82	NT	SB	SB	NT
MARIE LOUISE BACHIR	8,79	NT	SB	SB	NT
MIKHAIL SALEM	7,46	NT	NT	NT	NT
MUHAMADOU JIN	6,15	NT	NT	NT	AP
NATHALIE AHMED	6,16	NT	NT	NT	AP
OLEG KADDOUR	7,09	NT	NT	NT	AP
OMAR JIANG	8,99	NT	NT	NT	NT
OUSMAN HU	9,16	NT	NT	NT	NT
RASA ROBINSON	6,41	NT	SB	NT	AP
ROMEO JONES	8,77	NT	SB	SB	NT
SAIKOU QIU	6,92	NT	NT	NT	NT
SARA LIN	8,81	NT	NT	NT	NT
SARAH CHERIF	6,79	NT	NT	NT	AP
SOPHIE SAID	7,33	SB	NT	SB	NT
STEFANO JACKSON	9,71	NT	NT	NT	NT
VERONIQUE ALI	9,09	NT	NT	NT	NT
VIKTORIJA RODRIGUEZ	6,80	NT	NT	NT	NT

Nombre	Nota real	Prueba 1	Prueba 2	Prueba 3	Prueba 4
VLADIMIR HAMDI	9,53	NT	SB	AP	NT

❖ **Regresión lineal**

Nombre	Nota real	Prueba 1	Prueba 2	Prueba 3	Prueba 4
ABDOULIE SUN	8,29	SB	SB	SB	SB
AICHA LI	8,38	NT	NT	NT	NT
ALESSANDRO DAVIS	6,57	SB	SB	AP	NT
ALEXANDER BOURAS	7,81	SB	NT	AP	NT
ALEXEY AMARA	7,59	AP	AP	AP	AP
ANDREA MILLER	6,38	NT	NT	AP	AP
ANDREY ABED	7,76	NT	SB	AP	NT
ANNE MARIE MOHAMED	8,26	SB	SB	NT	NT
ANTONIO ANDERSON	8,85	NT	SB	NT	NT
ASMA JOHNSON	9,21	SB	SB	SB	SB
BAKARY JI	9,27	SB	SB	SB	NT
CAROLINE BRAHIM	8,96	NT	NT	NT	NT
CHANTAL MOHAMED SALEM	0,00	NT	AP	SS	SS
DAIVA CLARK	8,93	SB	SB	NT	SB
DMITRY HADDAD	9,68	NT	SB	SB	SB
FATIMA CHEN	4,84	NT	NT	SS	SS
GIUSEPPE MOORE	6,83	NT	NT	NT	AP
IGOR DJILALI	7,31	NT	SB	NT	NT
INGA MARTINEZ	7,74	NT	SB	NT	NT
INGRID LARBI	6,47	NT	NT	NT	NT
IVAN BENAISSA	8,73	NT	NT	NT	SB

Nombre	Nota real	Prueba 1	Prueba 2	Prueba 3	Prueba 4
JOLANTA LEWIS	7,19	NT	NT	NT	SB
JOSEPH BROWN	6,84	NT	SB	SB	NT
KARIMA LIU	7,23	NT	NT	NT	SB
KRISTINA HARRIS	8,15	NT	SB	SB	SB
LAMIN HUANG	5,71	NT	NT	AP	AP
LAURA GARCIA	6,61	SS	AP	SB	NT
LAURA SLIMANI	8,87	SB	SB	NT	NT
LINA MARTIN	8,71	SB	SB	NT	NT
LUCA TAYLOR	8,43	SB	SB	SB	SB
MANPREET SMITH	7,82	SB	SB	NT	NT
MARIE LOUISE BACHIR	8,79	NT	SB	AP	NT
MIKHAIL SALEM	7,46	SB	SB	NT	NT
MUHAMADOU JIN	6,15	SB	SB	NT	NT
NATHALIE AHMED	6,16	AP	NT	NT	NT
OLEG KADDOUR	7,09	SS	AP	NT	AP
OMAR JIANG	8,99	SB	SB	NT	NT
OUSMAN HU	9,16	NT	NT	NT	SB
RASA ROBINSON	6,41	SB	SB	NT	NT
ROMEO JONES	8,77	SB	SB	NT	SB
SAIKOU QIU	6,92	NT	NT	SB	NT
SARA LIN	8,81	NT	NT	NT	NT
SARAH CHERIF	6,79	NT	NT	SB	NT
SOPHIE SAID	7,33	NT	NT	AP	SS
STEFANO JACKSON	9,71	SB	SB	SB	SB
VERONIQUE ALI	9,09	NT	NT	SB	SB
VIKTORIJA RODRIGUEZ	6,80	NT	NT	SB	NT

Nombre	Nota real	Prueba 1	Prueba 2	Prueba 3	Prueba 4
VLADIMIR HAMDI	9,53	AP	AP	AP	NT

Bibliografía

[1] M. Á. Goberna Torrent y M. A. López Cerdá, «La predicción del rendimiento como criterio para el ingreso en la universidad,» *Revista de educación,* nº 283, pp. 235-248, 1987.

[2] F. Meneses Ponzini y J. Toro Cáceres, «Predicción de notas en Derecho de la Universidad de Chile: ¿sirve el ranking?,» *Revista_ ISEES,* nº 10, pp. 43-58, 2012.

[3] «Careers360 - The Education Hub,» 2017. [En línea]. Available: http://www.careers360.com/result-predictor. [Último acceso: Mayo 2017].

[4] «Yingwenhua,» 2017. [En línea]. Available: http://www.yingwenhua.net/2016/06/how-to-use-the-online-jee-main-rank-predictor-college-tool/. [Último acceso: Mayo 2017].

[5] M. V. García Jiménez, J. M. Alvarado Izquierdo y A. Jiménez Blanco, «La predicción del rendimiento académico: regresión lineal versus regresión logística,» *Psicothema,* vol. 12, nº Supl. 2, pp. 248-252, 2000.

[6] «FrontRow,» 2017. [En línea]. Available: https://www.frontrowed.com/. [Último acceso: Mayo 2017].

[7] Trello, «Trello,» 2017. [En línea]. Available: https://trello.com. [Último acceso: Marzo 2017].

[8] T. Peters, «Python,» Python Software Foundation, 2001-2017. [En línea]. Available: https://www.python.org/dev/peps/pep-0020/. [Último acceso: Mayo 2017].

[9] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly, 2012.

[10] F. Pedregosa, «Scikit-Learn: Machine Leaning in Python,» 2011. [En línea]. Available: http://scikit-learn.org/stable/. [Último acceso: Mayo 2017].

[11] «Higcharts,» 2017. [En línea]. Available: https://www.highcharts.com/. [Último acceso: Mayo 2017].

[12] A. Ronacher, «Flask,» 2010. [En línea]. Available: http://flask.pocoo.org/. [Último acceso: Abril 2017].

[13] A. Ronacher, «Jinga2,» 2008. [En línea]. Available: http://jinja.pocoo.org/docs/2.9/. [Último acceso: Abril 2017].

[14] D. Jacobs y H. Yang, «FlaskWTF,» 2010 - 2013. [En línea]. Available: https://flask-wtf.readthedocs.io/en/stable/. [Último acceso: Abril 2017].

[15] Sphinx, «WTForms,» WTForms Team, 2010. [En línea]. Available: https://wtforms.readthedocs.io/en/latest/. [Último acceso: Abril 2017].

[16] «myBalsamiq,» Balsamiq Studios, LLC, 2008-2017. [En línea]. Available: https://www.mybalsamiq.com/. [Último acceso: Marzo 2017].

[17] «MySQL,» OracleCorporation, 2017. [En línea]. [Último acceso: Marzo 2017].

[18] «MariaDB,» MariaDB Foundation, 2017. [En línea]. Available: https://mariadb.org/learn/. [Último acceso: Abril 2017].

[19] J. de la Horra Navarro, Estadística Aplicada, Ediciones Díaz de Santos, 2003.

[20] J. Wiley, Clustering Algorithms, 1975.

[21] «Scikit-Learn. Cluster,» scikit-learn developers, 2010-2016. [En línea]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans. [Último acceso: Mayo 2017].

[22] «Scikit-Learn. KMeans,» scikit-learn developers, 2010-2017. [En línea]. Available: http://scikit-learn.org/stable/modules/clustering.html#k-means. [Último acceso: Mayo 2017].

[23] Weston.pace, «Wikipedia,» 26 Julio 2013. [En línea]. Available: https://es.wikipedia.org/wiki/K-means#/media/File:K_Means_Example_Step_4.svg. [Último acceso: Mayo 2017].

[24] «Scikit-Learn,» scikit-learn developers, 2010-2016. [En línea]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans . [Último acceso: Mayo 2017].

[25] M. Garré, J. J. Cuadrado y M. Á. Sicilia, «Comparación de diferentes algoritmos de clustering en la estimaión de coste en el desarrollo de software,» Alcala de Henares.

[26] Sigwert, «Wikipedia,» 30 Septiembre 2011. [En línea]. [Último acceso: Mayo 2017].

[27] «Wikipedia,» 26 Enero 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Distancia_euclidiana. [Último acceso: Mayo 2017].

[28] «Wikipedia,» 25 Febrero 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Geometr%C3%ADa_del_taxista. [Último acceso: Mayo 2017].

[29] «SciPy.org,» The Scipy community, 2008-2016. [En línea]. Available: https://docs.scipy.org/doc/scipy/reference/. [Último acceso: Mayo 2017].

[30] S. Raschka, Python Machine Learning, Packt Publishing Ltd., 2015.

[31] «WIkipedia. Reglas de asociación,» 11 Mayo 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Reglas_de_asociaci%C3%B3n. [Último acceso: Mayo 2017].

[32] «Wikipedia,» 25 Abril 2017. [En línea]. Available: https://en.wikipedia.org/wiki/Association_rule_learning#Support. [Último acceso: Mayo 2017].

[33] «Scikit-Learn. Complejidad árboles de decisión,» 2010-2016. [En línea]. Available: http://scikit-learn.org/stable/modules/tree.html#complexity. [Último acceso: Mayo 2017].

[34] «Wikipedia. Impureza de Gini,» 2 Abril 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Aprendizaje_basado_en_%C3%A1rboles_de_decisi%C3%B3n#Impureza_de_Gini. [Último acceso: Mayo 2017].

[35] «Wikipedia. Regresión lineal,» 27 Abril 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal. [Último acceso: Mayo 2017].

[36] «Django,» Django Software Foundation, 2005-2007. [En línea]. Available: https://www.djangoproject.com/. [Último acceso: Marzo 2017].

[37] G. Dwyer, «Codementor,» Codementor Community, 13 Febrero 2017. [En línea]. Available: https://www.codementor.io/garethdwyer/flask-vs-django-why-flask-might-be-better-4xs7mdf8v. [Último acceso: Abril 2017].

[38] R. Brown, «AirPair,» 2015. [En línea]. Available: https://www.airpair.com/python/posts/django-flask-pyramid. [Último acceso: Abril 2017].

[39] D. Beazley y B. K. Jones, Python Cookbook, O'Reilly, 2013.

[40] «OpenTechSchool,» OpenTechSchool, 2013-2014. [En línea]. Available: http://esparta.github.io/python-data-intro/. [Último acceso: Octubre 2016].

[41] G. James, D. Witten, T. Hatie y R. Tibshirani, An Introduction to Statistical Learning with Applications in R, Nueva York: Springer , 2013.

[42] A. Urquía y C. Martín, «UNED,» [En línea]. Available: http://www.euclides.dia.uned.es/aurquia/Files/analisisDatosR.pdf. [Último acceso: Octurbre 2016].

[43] F. Tusell, «Universidad del País Vasco,» 2004-2005. [En línea]. Available: http://www.et.bs.ehu.es/~etptupaf/pub/papiros/s-demo3.pdf. [Último acceso: Octubre 2016].

[44] «Tutorial de R,» Escuela Andaluza de Salud Pública, [En línea]. Available: http://www.tutorialr.es/es/registrese.cfm. [Último acceso: Octubre 2016].

[45] M. Tirados, «Big Data Hispano,» 25 Mayo 2015. [En línea]. [Último acceso: Octubre 2016].

[46] Jim.belk, «Wikipedia,» 23 Septiembre 2007. [En línea]. [Último acceso: Mayo 2017].

[47] Psychonaut, «Wikipedia,» 25 Abril 2006. [En línea]. [Último acceso: Mayo 2017].